

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

A COST AND UTILITY ANALYSIS OF NIM/CAMAC STANDARDS AND EQUIPMENT FOR SHUTTLE PAYLOAD DATA ACQUISITION AND CONTROL SYSTEMS

VOLUME III. TASKS 3 & 4

30 JUNE 1976

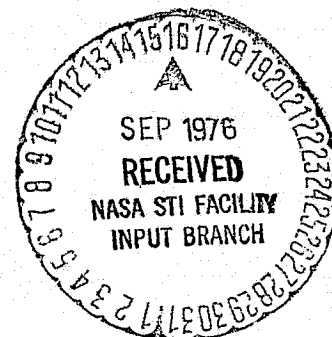
(NASA-CR-147854) A COST AND UTILITY
ANALYSIS OF NIM/CAMAC STANDARDS AND
EQUIPMENT FOR SHUTTLE PAYLOAD DATA
ACQUISITION AND CONTROL SYSTEMS. VOLUME 3:
TASKS 3 AND 4 (TRW Systems Group) 145 p HC G3/19

N76-31280
HC \$4.50

Unclas
50442

Prepared for
LYNDON B. JOHNSON SPACE CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
HOUSTON, TEXAS 77058

CONTRACT NAS9-14693



TRW
DEFENSE AND SPACE SYSTEMS GROUP

ONE SPACE PARK • REDONDO BEACH • CALIFORNIA

A COST AND UTILITY ANALYSIS OF NIM/CAMAC STANDARDS AND EQUIPMENT FOR SHUTTLE PAYLOAD DATA ACQUISITION AND CONTROL SYSTEMS

VOLUME III. TASKS 3 & 4

30 JUNE 1976

Prepared for
LYNDON B. JOHNSON SPACE CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
HOUSTON, TEXAS 77058

CONTRACT NAS9-14693

TRW
DEFENSE AND SPACE SYSTEMS GROUP

ONE SPACE PARK • REDONDO BEACH • CALIFORNIA

FOREWORD

A Cost and Utility Analysis of NIM/CAMAC Standards and Equipment for Shuttle Payload Data Acquisition and Control Systems was performed by the Defense and Space Systems Group of TRW, Inc. under Contract NAS9-14693 for the Lyndon B. Johnson Space Center of the National Aeronautics and Space Administration. The work was managed by Dr. Richard J. Kurz (Telephone (213) 535-2936) of the Instrument Systems Department, TRW Defense and Space Systems Group. The study was administered under the technical direction of Dr. Richard D. Eandi (Telephone (713) 483-5176) of the Space Physics Branch, Johnson Space Center.

The results of the study are presented in three volumes:

VOLUME I. SUMMARY

Overall summary of the analyses and conclusions

VOLUME II. TASKS 1 AND 2

Identification and selection of representative payloads for analysis and functional analysis of the selected payloads for NIM/CAMAC equipment applicability and commonality.

VOLUME III. TASKS 3 AND 4

Analysis of the modifications to NIM/CAMAC equipment required for compatibility with the Spacelab environment and their estimated cost, development of a management plan for the utilization of NIM/CAMAC equipment and programmatic cost estimates, and assessment of the implementation and impact of CAMAC software.

CONTENTS

	Page
1. MODIFICATION ANALYSIS OF NIM/CAMAC EQUIPMENT (TASK 3)	1
1.1 Introduction	1
1.2 Analysis of NIM/CAMAC and Spacelab Specification (Task 3A - First Phase)	2
1.2.1 NIM/CAMAC Equipment Characteristics and Specifications	2
1.2.2 Spacelab Environmental Requirements	7
1.2.3 Preliminary Assessment of NIM/CAMAC Compatibility with Spacelab Environments	13
1.2.4 Recommendations for the SECT Program	16
1.3 Analysis of NIM/CAMAC Suitability for Spacelab Environments (Task 3A - Second Phase)	19
1.3.1 Dynamic/Structural Analysis	19
1.3.2 Thermal Analysis	25
1.4 Recommended Modifications and Estimated Costs (Task 3B)	43
1.4.1 Recommended Modifications	43
1.4.2 Modification Costs	46
1.4.3 Analysis of Alternative Modified Equipment Sources	51
2. IMPLEMENTATION ANALYSIS AND MANAGEMENT PLAN DEVELOPMENT FOR NIM/CAMAC EQUIPMENT (TASK 4A)	53
2.1 Introduction	53
2.2 Time-Phased NIM/CAMAC Equipment Requirements	56
2.2.1 Baseline Spacelab Flight Traffic Model	56
2.2.2 Payload Equipment Usage Schedule	57
2.2.3 NIM/CAMAC Equipment Usage Projections	61
2.3 NIM/CAMAC Equipment Pool Analysis	63
2.3.1 Alternative Pool Concepts	64
2.3.2 Recommended Pool Concept	66
2.4 Equipment Costs and Pool Management Plan	68
2.4.1 Pool Equipment Procurement Requirements	68
2.4.2 Cost Estimate for Pool Equipment	71
2.5 Dedicated Equipment Approaches	74
2.5.1 Equipment Requirements for Dedicated Approaches	74
2.5.2 Estimated Costs for Dedicated Approaches	76
2.5.3 Comparison of Costs for the Alternative Approaches	78

CONTENTS (Continued)

Page

3.	IMPLEMENTATION AND IMPACT OF CAMAC ON SPACELAB EXPERIMENT SOFTWARE (TASK 4B)	80
3.1	Introduction	80
3.1.1	Scope of the Software Task	80
3.1.2	General Experiment Software System Requirements	81
3.1.3	Impact of CAMAC on Software Requirements	82
3.2	Existing Software Systems	85
3.2.1	Hot Fuel Examination Facility	85
3.2.2	Fermi National Accelerator Laboratory	89
3.2.3	Los Alamos Scientific Laboratory	93
3.2.4	Aluminum Company of America (ALCOA)	96
3.3	Spacelab Software System	99
3.4	CAMAC Support Software for Spacelab	102
3.4.1	Software for Facility Use of CAMAC	102
3.4.2	Software for Experiment Use of CAMAC	102
3.5	Software Requirements for Representative Payloads	105
3.5.1	Shuttle Infrared Telescope Facility Software	105
3.5.2	X-Ray/Gamma-Ray Pallet Payload	115

APPENDIXES

I	CAMAC SOFTWARE PRODUCTS GUIDE	I-1
II	SPACELAB SOFTWARE DESCRIPTION	II-1

1. MODIFICATION ANALYSIS OF NIM/CAMAC EQUIPMENT (TASK 3)

1.1 INTRODUCTION

Our objectives in this task were: 1) to determine what modifications NIM/CAMAC equipment in its current form, i.e., designed for ground-based laboratory use, would be required to permit its use in the Spacelab environment, and 2) to estimate the cost of these modifications and identify the most cost-effective approach to implementing them. Our effort was correspondingly divided into two tasks, the first of which was performed in two phases.

Task 3A (First Phase) - Assemble and Evaluate NIM/CAMAC and Spacelab Information

- Compile available NIM/CAMAC data and specifications.
- Review various Spacelab specifications and update the Rockwell Spacelab/Experiment Equipment Interface Requirements (SEEIR) to recommend design criteria.
- Assess incompatibilities between existing NIM/CAMAC equipment and current Spacelab requirements.
- Determine needs for additional data on NIM/CAMAC equipment.
- Prepare preliminary recommendations to the Shuttle Environmental Compatibility Test (SECT) program.

Task 3A (Second Phase) - Analyze NIM/CAMAC Suitability for Spacelab Environments

- Analyze NIM/CAMAC equipment with respect to dynamic, thermal, electromagnetic compatibility; and parts, materials, and processes characteristics.
- Prepare recommendations for SECT based on analytical results.
- Review SECT results.

Task 3B - Analyze Required Modifications and Determine Costs

- Determine NIM/CAMAC modifications required to meet Spacelab environments.
- Estimate modification costs.
- Identify cost-effective modification source.

Due to its standardized nature, NIM/CAMAC equipment has a considerable amount of inherent mechanical commonality, irrespective of manufacturer or function. This fact has allowed us to perform much of the modification

analysis in parallel with the functional analysis carried out in Task 2 of the study. In addition, the greater applicability of CAMAC equipment found in the course of performing Task 2 led us to place a greater emphasis on CAMAC equipment in Task 3.

Because of delays in the SECT program, which was intended to be carried out in parallel by JSC, test data were not available during the contract period of performance for review as originally planned in the second phase of Task 3A. As a consequence, we undertook a more extensive dynamic and structural analysis than had originally been intended. Again, the standardized nature of NIM/CAMAC equipment made it meaningful to perform detailed analysis using a generalized structural computer model of the equipment.

Our original planning for Task 3B included an analysis of the trade-off between the degree and costs of equipment modification and changes to the cost-driving Spacelab environmental requirements. As we will see from the results of Task 3 and the programmatic cost estimation in Task 4A (see Section 2), the costs involved in even the most extensive equipment modifications that we will consider are relatively small on the scale of the costs that would be involved in making significant changes to the important Spacelab environments such as random vibration. We believe, therefore, that it is clearly more cost effective to modify NIM/CAMAC equipment to be compatible with the Spacelab environment rather than the converse.

The basic reference documents used in performing Task 3 are listed in Table 1-1. A new version, May 1976, of the Spacelab Payload Accommodation Handbook has recently become available, but the differences between the 1975 and 1976 versions do not significantly affect the results of this study. Although they are not listed in Table 1-1, numerous catalogs and specification sheets from NIM and CAMAC equipment manufacturers were also used in addition to the publications listed in Table 1-1 of Volume II.

1.2 ANALYSIS OF NIM/CAMAC AND SPACELAB SPECIFICATIONS (TASK 3A - FIRST PHASE)

1.2.1 NIM/CAMAC Equipment Characteristics and Specifications

The overall physical characteristics of NIM and CAMAC equipment are controlled by the standards. Specification drawings for both systems are available. Drawings for CAMAC are contained in ERDA Report TID-25875 and a

Table 1-1. References Used in the NIM/CAMAC Equipment Modification Analysis

Spacelab Payload Accommodations Handbook, ESRO, May 1975.

Space Shuttle System Payload Accommodation, JSC 07700, Volume XIV, Revision D, November 1975.

Analysis of Commercial Equipment and Instrumentation for Spacelab Payloads, Rockwell SD74-SA-0047, September 1974.

Feasibility Study of Common Electronic Equipment for Shuttle Sortie Experiment Payloads, Bendix BSR 4142, June 1974.

Natural and Induced Environments, ERNO SR-ER-0008, February 1976.

Random Vibration Flight Environments at Spacelab Equipment Locations, ERNO TN-ER-40-020-75, October 1975.

Engineering Development Test Procedures for NIM/CAMAC Instrumentation, NASA/JSC, March 1976.

Selected publications regarding NIM and CAMAC, see Table 1-1, Volume II of this report.

drawing set (CAPE-1189) for NIM is available from the Clearinghouse for Federal Scientific and Technical Information. The general physical characteristics of the CAMAC system can be seen in Figures 1-1 through 1-4. Although the construction details vary between models and manufacturers, the modules shown in Figure 1-3 and 1-4, with side covers removed, are reasonably representative of typical CAMAC and NIM, respectively.

The NIM and CAMAC standards do not specify environmental requirements other than a general statement that the equipment is intended for use in environments typically associated with laboratory instrumentation (e.g., the ambient temperature range of roughly 0 °C to 50 °C). Very little published data on environmental characteristics of NIM and CAMAC equipment exist. The CERN laboratory of the European Organization for Nuclear Research has generated a series of internal test reports on NIM and CAMAC modules that include some thermal test results. Typically, individual modules are checked for proper functional performance over the temperature range of 0 to 60 °C. In several cases, vibration tests were also performed but the test conditions are not defined in the reports.

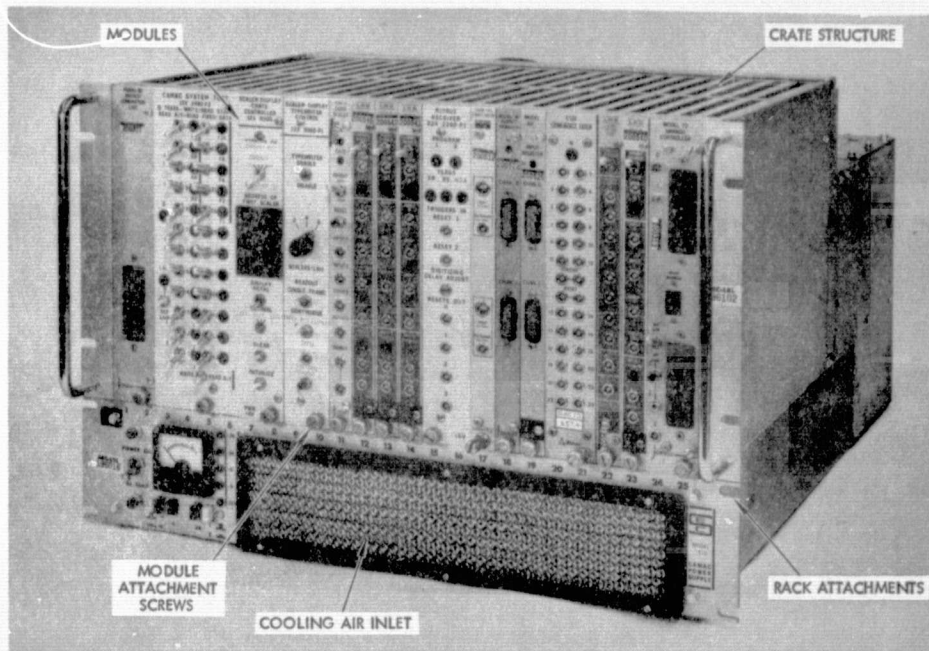


Figure 1-1. Front View of CAMAC Modules Loaded in a CAMAC Crate

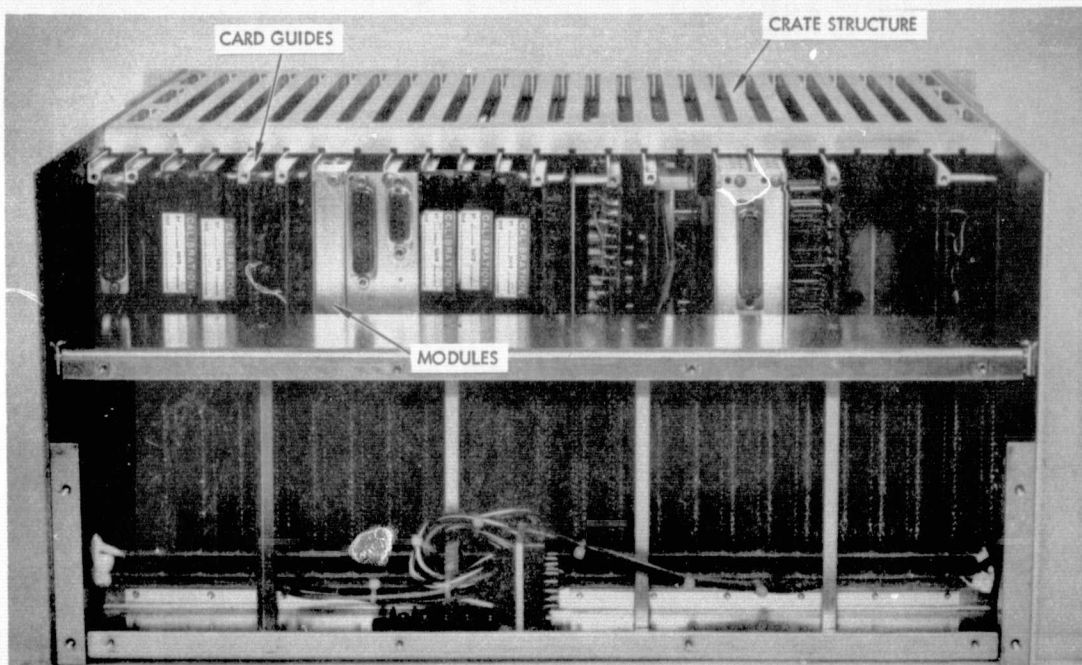


Figure 1-2. Rear View of CAMAC Crate and Modules with Power Supply Removed

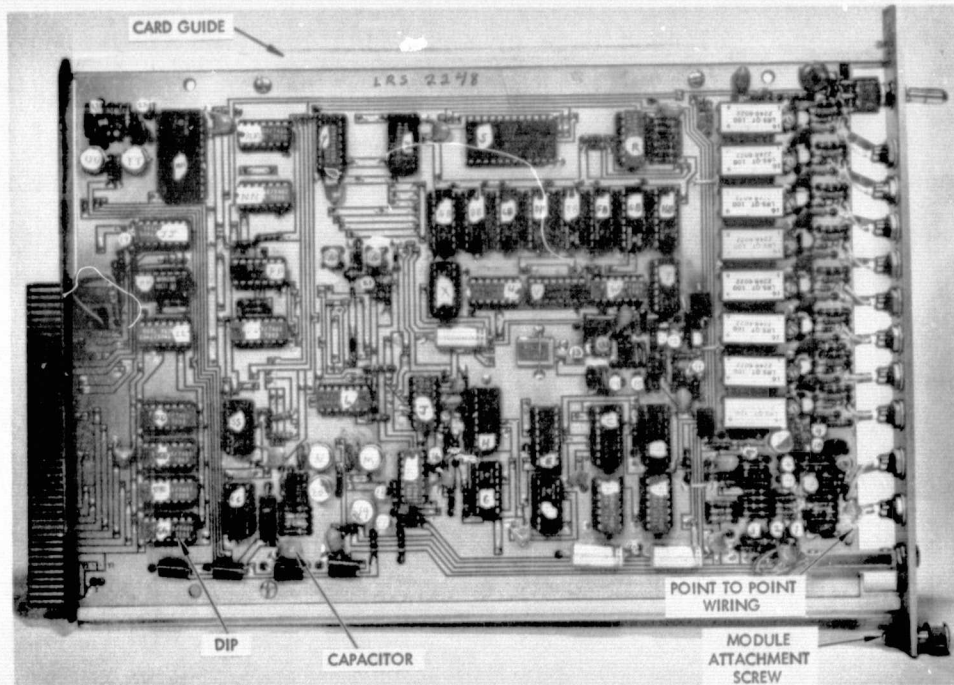


Figure 1-3. Typical CAMAC Module with Side Cover Removed

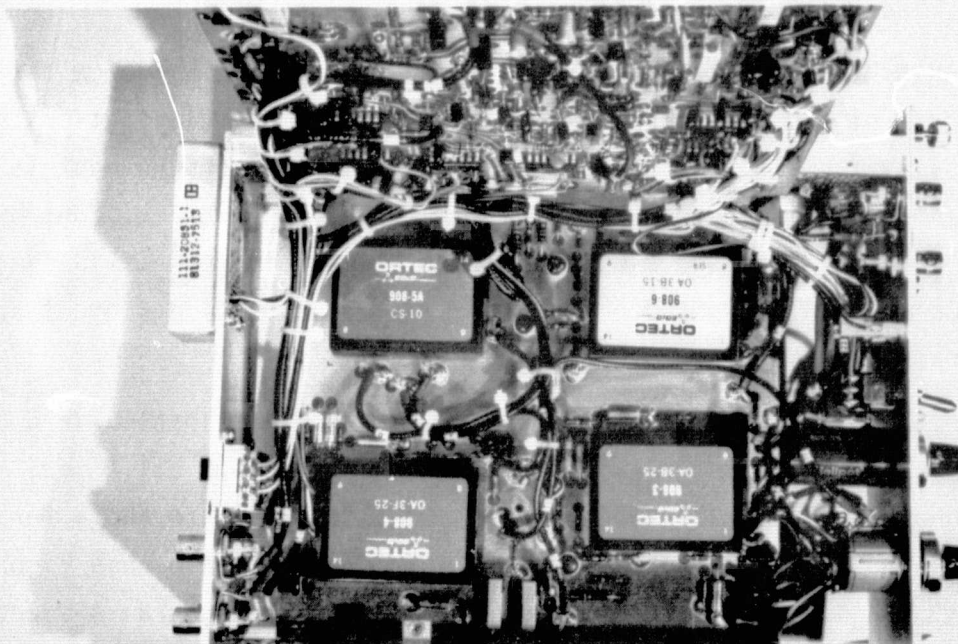


Figure 1-4. Typical NIM Module with Side Cover Removed

In lieu of any published data or specifications, we contacted a number of equipment manufacturers as well as equipment users. In particular, a trip was made to Lawrence Berkeley Laboratory of the University of California where a large amount of NIM and CAMAC equipment is used and some special purpose NIM and CAMAC equipment is designed and fabricated for internal use. It was possible to both inspect a wide spectrum of NIM and CAMAC equipment from a number of manufacturers and to discuss operating experience with equipment users. In addition, several engineers at LBL are members of the NIM Committee and have been involved in the standardization activity for both NIM and CAMAC since its inception.

The results of this information gathering that are relevant to the question of NIM/CAMAC usage in Spacelab payloads are summarized in the following sections.

1.2.1.1 Structural and Dynamic Properties

Although the structural characteristics of NIM and CAMAC equipment are reasonably well defined by the standards, and the manufacturers uniformly conform to the standards, essentially no formal analysis or testing of the structural behavior under dynamic environments have been conducted.

1.2.1.2 Thermal Characteristics

Both NIM and CAMAC equipment depend on convective air flow for cooling. In the case of NIM equipment, where the maximum power dissipation in a bin is typically 72 watts, natural convection in a one-g environment is adequate for reliable operation at ambient temperatures up to 50 °C. Fans are necessary to provide increased air flow when either the power dissipation in an individual bin is increased or when a number of bins of equipment are stacked in one rack enclosure.

For CAMAC equipment, where the maximum power dissipation in a crate is typically 300 watts, forced-air cooling must be used at all times. The CAMAC standards recommend at least 48 cfm of air-flow per crate and a forced-air system is routinely included as part of the powered crate (see Figures 1-1 and 1-2). Fans, located in a plenum below the modules, draw ambient air through a front panel filter and distribute the air up through the modules. The actual air flow provided in commercial crates is not specified. The

manufacturers use whatever fan capacity is required to obtain reliable operation in the field. For example, one of the most commonly used crates, manufactured by Standard Engineering Corporation, uses three fans that have a total free-air capacity of 350 cfm.

1.2.1.3 Parts, Materials, and Processes Characteristics

The basic modules consist of a printed-wiring circuit board attached to a metal frame (see Figures 1-3 and 1-4). The electronic parts used by NIM and CAMAC manufacturers are almost universally industrial grade, 0 °C to 70 °C operating range. Next to the required functional characteristics, cost is usually the most important factor in parts selection. The frequently used parts are of the type that are generally available in full military temperature range (-55 °C to 125 °C) and high-reliability versions.

The types of connectors that can be used on NIM and CAMAC equipment are defined by the standards. 50-ohm BNC type or NIM-CAMAC-Type 50-CM coaxial connectors are specified. The CAMAC dataway connector is an 86-contact card edge connector specified in the CAMAC standards. The standard NIM power connector is a special multicontact, mating pin-socket connector that includes guide pins and sockets. The CAMAC standards recommend two types of multicontact mating rectangular connectors for auxiliary use: 52-contact 2D subminiature, and 88- or 152-contact WSS subminiature.

The types of materials used in NIM and CAMAC equipment are quite common. Structural elements are aluminum, printed circuit boards are epoxy fiberglass and wiring is multistrand, Teflon or PVC insulated. Subminiature RG-type coaxial cable is frequently used for noise sensitive signal runs inside the modules. Miscellaneous uncontrolled plastics are present in the form of knobs and component cases.

Although the standards do not control manufacturing workmanship and methods or specify any quality assurance requirements, the commercial modules do conform uniformly to good standard commercial practice. The main process of concern in module fabrication is soldering, and the general quality of workmanship observed was good. With respect to wiring, typically very little stress relief or mechanical support to wiring is used.

1.2.2 Spacelab Environmental Requirements

For the analysis of NIM/CAMAC equipment, we were primarily concerned with the environmental requirements on equipment mounted in the racks of the

Spacelab module since this is the most likely location for NIM/CAMAC equipment. Other possible locations for the equipment include rack-mounted on the rear deck of the Orbiter cabin, mounted in a Spacelab Igloo, or pallet-mounted in the payload bay. The Orbiter cabin environments are not significantly different from the Spacelab module environments. The Igloos are currently intended to be used only for Spacelab subsystem equipment. If Igloos are available for experiment equipment, there is still at least one significant difference in the environment from the standpoint of NIM/CAMAC equipment. Although the Igloo is pressurized and temperature controlled, no forced-air convective cooling capability is planned. It is possible to consider modification of NIM/CAMAC equipment to allow operation without convective cooling and the NASA/GSFC NIM/CAMAC activity is, in fact, doing so. Although we will discuss this possibility briefly in our assessment of NIM/CAMAC compatibility with the Shuttle environments, our primary attention will be directed to the Spacelab module environment.

Assuming the equipment is located in the Spacelab module, the environments that are of principal interest are the dynamic and thermal environments. At the time of our activity to establish environmental requirements for Spacelab rack-mounted equipment, several different environmental specifications were available as design criteria. The appropriate specifications contained in the first four references listed in Table 1-1 are compared in Table 1-2 and the specification recommended for use in this analysis is defined.

The recommended specification was an extrapolation of our own spacecraft experience, a comparison of acoustic levels, and an examination of the shielding provided by the Shuttle and Spacelab. NASA/JSC, for their SECT program, increased the proposed random vibration environment to an overall test level of 12 grms, and we consequently performed our dynamic analyses using this level. The more recent references listed in Table 1-1 indicate that equipment will be exposed to an even more benign environment than that recommended by TRW in Table 1-2. The random vibration overall level is down to 3.3 grms in SR-ER-0008 and the May 1976 Spacelab Payload Accommodations Handbook. We expect that when complete Shuttle tests have been conducted, the more benign environment will prove to be correct and our calculations and the JSC tests will generally have been conservative.

Table 1-2. Comparison of Shuttle Payload Environments

	SPACELAB	ROCKWELL	BENDIX	VOL XIV	TRW RECOMMENDATION
Sinusoidal Vibration	5-35 Hz 0.25 g	x-axis 3-8.5 Hz (0.8" DA) 8.5-35 Hz (3.0 g) 35-50 Hz (1.0 g) y- and z-axis 3-7 Hz (0.8" DA) 7-35 Hz (2.0 g)	Undefined	5-35 Hz (0.25 g)	Nonrandom vibration would control in frequency-sensitive region (>50 Hz). A 1.0-g Sine Sweep from 10-2000 Hz at 1.0-Octave/Min is recommended for determination of dynamic characteristics
Random Vibration	20-200 Hz (+8 dB/oct) 200-700 Hz (0.1 g ² /Hz) 700-900 Hz (-18 dB/oct) 900-2000 Hz (0.02 g ² /Hz) Overall - 10-g rms Duration Undefined	20-60 Hz (+6 dB/oct) 60-500 Hz (0.14 g ² /Hz) 500-2000 Hz (-9 dB/oct) Overall - 10.6-g rms Duration Undefined	Shape Undefined Probably Flat from 70-130 Hz Overall - 14-g rms Duration Undefined	50-110 Hz (+6 dB/oct) 110-700 Hz (0.9 g ² /Hz) 700-1200 Hz (-9 dB/oct) Undefined	20-200 Hz (+8 dB/oct) 200-700 Hz (0.1 g ² /Hz) 700-900 Hz (-18 dB/oct) 900-2000 Hz (0.02 g ² /Hz) Overall - 10-g rms Duration - 1 minute

Table 1-2. Comparison of Shuttle Payload Environments (continued)

	SPACELAB	ROCKWELL	BENDIX	VOL XIV	TRW RECOMMENDATION
Acoustics					
Overall	138 dB	Undefined	Undefined	145 dB	None - random vibration is adequate to envelope effects of acoustics.
Max Level (1/3 Octave)	130.5 dB	131 dB		135 dB	
Frequency for Max Level	200-500 Hz	300-400 Hz		200-500 Hz	
Pyrotechnic Shock	Undefined	Undefined (Will be less than vibration/acceleration)	Undefined	Undefined	None - considering the location of the equipment, shock would be attenuated to nondestructive levels.
Landing	1.5 g 260 millisecc Rectangular	Undefined (Will be less than vibration/acceleration)	Undefined	1.5 g 260 millisecc Rectangular	None - covered by vibration.
Crash Shock					
Level	40 g	9 g	8 g	40 g	None - covered by vibration for internal structure.
Duration	11 millisecc	Undefined	Undefined	11 millisecc	
Pulse Shape	Sawtooth			Sawtooth	Should only be used for analysis of mounting hardware to ensure no catastrophic failure
Constant Acceleration (Worst Case)	4.0 g (-x)	3.3 g (-x)	<u>±</u> 4 g all axes	4.4 g (x)	None - covered by vibration.

Table 1-2. Comparison of Shuttle Payload Environments (continued)

	SPACELAB	ROCKWELL	BENDIX	VOL XIV	TRW RECOMMENDATION
Temperature (Air Cooled)					
Max Inlet Temp.	35 °C	29 °C	29 °C	N/A	35 °C
Min Inlet Temp.	Undefined	26 °C	18 °C		Not required
Pressure	1.0 Bar	14.7 psia	14.7 psia		14.7 psia
Flow Rate	.22 kg/hr/watt	Undefined	Undefined		.22 kg/hr/watt

Our recommended general structural and thermal analysis criteria are presented in Table 1-3.

Table 1-3. Structural and Thermal Analysis Criteria

Structural

- The random vibration environment will be controlling since it will excite resonances causing inertial loads exceeding those of sinusoidal vibration, acoustics, pyrotechnic shock, and landing shock for most structures. Crash shock should be considered for mounting hardware only.
- Load factors for analysis will be based upon 3-sigma resonance response of a single degree of a system having a transmissibility of ten.
- A minimum resonant frequency is to be determined to preclude peak circuit board deflections in excess of 0.1 inch, double amplitude.
- Margins of safety for structural members should be at least 1.25, based upon yield strength.

Thermal

- Forced-air cooling will be available with a maximum inlet temperature of 35 °C and a maximum flow rate of 0.22 kg/hr per watt of power dissipated in the equipment.
- Allowable electronic part operating temperatures should not exceed 125 °C maximum and 70 °C preferred.

1.2.3 Preliminary Assessment of NIM/CAMAC Compatibility with Spacelab Environments

1.2.3.1 Dynamic Environments

Based on experience with space electronics, our overall assessment after examination of a variety of NIM/CAMAC equipment is that it should be capable of surviving the dynamic environments without major structural modification. The determination of the detailed modifications or changes needed will require analysis and/or well-instrumented vibration testing.

The modifications that we thought would be required are:

Attachment of Crates or Bins to Racks - The normal front panel attachments of the crate and bins to a rack are inadequate. Front attachments by themselves result in a large cantilevered load at a weak section of the front panel. The structure must be modified to a rail mount that is compatible with Spacelab rack design.

Module Attachment to the Crate - Modules must have top and bottom screws installed with a controlled torque rather than just one thumb screw as now used typically. Also, spring fingers or other retaining devices should be placed in the card guides to eliminate guide-to-card rail clearance.

Module Modifications - The corrective actions listed below should be taken to prevent structural failures at the component level.

- Conformal coating is not used on the circuit boards. Thin conformal coating (3 to 5 mils) should be used to provide mechanical support for small axial lead parts and to provide some vibration damping.
- Epoxy bond parts of five grams or heavier not having other support.
- Point-to-point wiring that is not stress relieved at points of relative motion (such as from front of module to card) is susceptible to fatigue failure. These wires should be routed with slack, and a bond or other means of support provided near solder joints.
- Some axial lead parts have lead bends quite close to the part body that can possibly result in vibration failure of the leads. Manufacturing guidelines to preclude lead bending at the part body should be imposed.
- Some parts, such as disc capacitors, are mounted vertically without mechanical support for the part body. These should be laid down and bonded or otherwise supported.
- CAMAC card edge plug-in connectors are not flight qualified and may be a vibration problem. Some structural support such as guide pins should be provided.

- Printed circuit boards are quite large compared to most flight equipment. For flight, some cards may require the addition of stiffeners.
- Integrated circuits in dual in-line packages (DIP's) are used extensively. Problems have been experienced with DIP's failing after environmental exposure. Manufacturing guidelines on lead forming and part installation should be imposed to minimize installation stresses at the DIP body.
- Fasteners are not locked. To preclude loosening during vibration, torques should be specified and positive locking mechanisms provided such as locking hardware or epoxy bonds.

1.2.3.2 Thermal Environments

Our general assessment of the compatibility of NIM/CAMAC equipment with the Spacelab thermal environment is that the available forced-air convective cooling capacity is marginal. Operation without forced-air cooling will not be possible without significant modifications to reduce power consumption and improve the conductive heat paths in the equipment.

To be compatible with the Spacelab module forced-air cooling system for rack-mounted equipment, a plenum, which provides connection between a crate or bin and the rack air return ducts, will be required. This plenum should also be designed to provide uniform air flow over all of the modules mounted in the crate or bin.

Even assuming such an arrangement is provided, the situation is marginal. The Spacelab air flow rate of 0.22 kg/hr/watt corresponds to 32 cfm for a 300-watt crate. This is significantly lower than the 48 cfm recommended in the CAMAC standard, which itself may be below the flow rate used in commercial crates. Analysis and testing are definitely required to determine the degree of compatibility. The use of electronic parts capable of operating up to 125 °C is probably desirable in any case.

A very simplified calculation of the heat paths available to conduct heat from the electronic parts in a CAMAC module to the crate structure indicates that the module part temperatures will rise roughly 25 °C above the crate structure temperature for each watt of power dissipated in the module. Since the average power dissipation in commercial CAMAC modules is close to ten watts, conductive cooling is not adequate. Even with military temperature range parts, module power dissipations in excess of about four watts will not be tolerable without significant mechanical changes to improve the conductive heat paths available.

1.2.3.3 Electromagnetic Compatibility

Although it is very difficult to judge without actual test data, we believe that no significant problems should be encountered with electromagnetic compatibility. In general laboratory usage, electromagnetic compatibility problems within NIM/CAMAC systems or between NIM/CAMAC and other equipment seldom occur even though the application is frequently sensitive to electrical noise. The basic construction of the NIM and CAMAC system provides reasonably good (but certainly not complete) shielding of radiated emissions. The question of conducted emissions is more problematical. Although the standards provide for separate high-quality grounds and power return lines, etc., commercial NIM/CAMAC equipment usually has only one common circuit ground which is also tied to frame ground. The grounding practices used in commercial equipment will have to be improved by consistently maintaining isolation between circuit and frame grounds in order to preclude problems in meeting Spacelab EM compatibility requirements. EMC testing should be performed since it is very difficult to perform conclusive analysis of the problem.

1.2.3.4 Parts, Materials, and Processes

Parts - As was previously discussed in Section 1.2.3.2, the thermal environment may require the use of parts capable of operating at case temperatures up to 125 °C. In other words, military grade parts would have to be used in place of industrial grades. From the standpoint of quality assurance requirements on electronic parts, the use of high-reliability parts may not be necessary. NIM/CAMAC manufacturers frequently burn-in the active parts used in commercial units and perform elevated temperature acceptance testing of the equipment to eliminate defective components. At a minimum, these types of screening activities should be universally adopted for NIM/CAMAC equipment to be used in flight applications.

The types of coaxial connectors used in NIM/CAMAC equipment are designed for easy and convenient connect/disconnect. Users have experienced occasional problems with the reliability of the NIM-CAMAC-type 50 CM. Substitution of a space-qualified miniature 50-ohm connector type such as the Microdot S-50 series is recommended. The CAMAC dataway card edge connector is an open question. If circuit operation during the launch environments is required, the

use of card edge connectors should be eliminated. However, it appears that very little, if any, of the payload NIM/CAMAC equipment needs to be operated during launch. Thus, we are not convinced that the card edge connector is not acceptable. The question should be settled by testing.

Materials - NIM/CAMAC equipment should present very few problems since the basic materials used are the same as those frequently used in space electronics. Some material substitutions will have to be made to assure consistent use of flame retardant, low-volatility plastics. For example, only Teflon-insulated wire should be used. No particular outgassing problems are expected, especially if conformal coating is used as recommended in Section 1.2.3.1. Naturally, a low outgassing conformal coating, such as Solithane, should be selected. The same is true for epoxy bonding materials.

Processes - In view of the generally good quality of soldering observed in the sample of NIM/CAMAC equipment inspected, strict imposition of NASA soldering standards is probably not absolutely necessary. However, some criteria should be adopted to assure a consistently high level of workmanship. The recommendations made in Section 1.2.3.1 with regard to part lead forming and installation practices as well as wiring stress relief are at least as important as the soldering techniques.

1.2.4 Recommendations for the SECT Program

Initial recommendations for the SECT program were presented to JSC at the first briefing and review meeting. These recommendations were based upon the activities described up to this point and dealt with dynamic and thermal testing as the highest priority tests.

1.2.4.1 Dynamic Tests

Sinusoidal vibration testing was recommended to determine frequencies and amplifications. The test results are to be used to check the analytical results as well as to assist in understanding the system dynamics. Random vibration tests were recommended to simulate the expected flight environment in order to identify possible failure modes and to determine the peak random vibration inertial loads at critical points of the system. No shock, acoustic or constant acceleration tests were recommended because random vibration would present more severe loads.

Test Levels -

- Sinusoidal Vibration

1 g, 10-2000 Hz at 1 oct/min - each of 3 axes

● Random Vibration	<u>Frequency (Hz)</u>	<u>Level</u>
	20-200	+8 dB/octave
	200-700	0.1 g ² /Hz
	700-900	-18 dB/octave
	900-2000	0.02 g ² /Hz
	Overall Level	= 10-g rms
	Duration	= 1 minute per axis - each of 3 axes

Test Configuration -

- Each crate should have supporting devices to prevent cantilevering of the crate as is currently done commercially.
- Modules should have screws at top and bottom for torquing into crate structure.
- Card restraint should be used in crate card guide slots.
- Selected boards should be conformally coated so that identical boards with and without conformal coat can be compared.
- Guide pins should be provided at the CAMAC dataway connector for selected modules for comparison to identical modules without guide pins.
- Structural support frames should be provided for selected boards for comparison to identical boards without frames.

Test Instrumentation - Accelerometers for sinusoidal and random vibration should be placed at the locations below in the axis of excitation. The number of accelerometers has been limited to eleven per axis since this is typically the limit per tape recorder.

- # 1 to 6 - on PC cards center of each of 3 types with and without modification.
- # 7 - on one PC card near connector.
- # 8 - on one PC card center edge near front.
- # 9 - crate structure top center rear.
- #10 - crate structure bottom center rear.
- #11 - crate structure top edge rear.

Data Reduction -

- Sinusoidal Test - Filtered transmissibility plots of each of the in-line accelerometers for each of the three axes of tests.
- Random Vibration Tests - Power spectral density plots of each in-line accelerometer and very fast speed (~ 20 in/sec) chart recorder plots of g vs time for selected accelerometers having the highest response. Data to be used to pick highest g level for inertial loads.

1.2.4.2 Thermal Tests

The recommended thermal testing was intended to simulate the Spacelab forced-air cooling system. The thermocouple and air flow measurements should be made at locations that will provide results that can be used to check and update our thermal analyses as well as at locations that are expected to be the worst-case points.

Test Levels -

- Air Temperature - 35°C
- Flow Rate - .22 kg/hr/watt (.11 cfm/watt) and 25%, 50%, 150%, and 200% of the design rate.

Test Configuration -

- In simulated rack with full load of equipment.
- Closed Air Flow

Thermocouple Measurements -

- Inlet and Exit Air
- Critical Board or Part Temperatures

Air Flow Measurements: Pressure and Velocity -

- At Inlet and Exit
- At critical locations such as corners to evaluate flow patterns.

1.3 ANALYSIS OF NIM/CAMAC SUITABILITY FOR SPACELAB ENVIRONMENTS (TASK 3A - SECOND PHASE)

1.3.1 Dynamic/Structural Analysis

The detailed dynamic/structural analysis was undertaken to establish the capability of NIM/CAMAC equipment to withstand the Spacelab environments. Complete documentation of this analysis is contained in TRW Report 7517.2-854, "CAMAC-Dynamic Structural Analysis," April 18, 1976. As discussed in the previous section, it is clear that random vibration will be the structural design driver, and that a design that can withstand vibration loads will be compatible with the shock, acceleration, and acoustics environments.

Our vibration analysis was directed at obtaining resonant frequencies and mode shapes so that the dynamic stresses on structural elements of crate and modules could be calculated. In addition, printed circuit board deflections and peak accelerations in the system are required to evaluate the probability of failures at the component level.

1.3.1.1 Random Vibration Environment

The random vibration power spectral density used as the dynamic environment is shown in Figure 1-5 and is identical to the test levels to be used in the SECT program.

1.3.1.2 Crate Dynamic Model

A simplified drawing of the crate dynamic model used in the computer analysis is shown in Figure 1-6. The dynamic model assumes that the front upper and lower card guide castings are held together with three module front beams. This simulates the actual condition of multiple module front beams.

The only physical tie between the modules and crate that affects the crate structure is the module front panel which is attached to the crate by an upper and lower fastener. The weight of the modules is distributed to the crate node points along the card guide locations. The model assumes all module guide slots are filled, and that the modules are free to slide in one dimension in the guide slots.

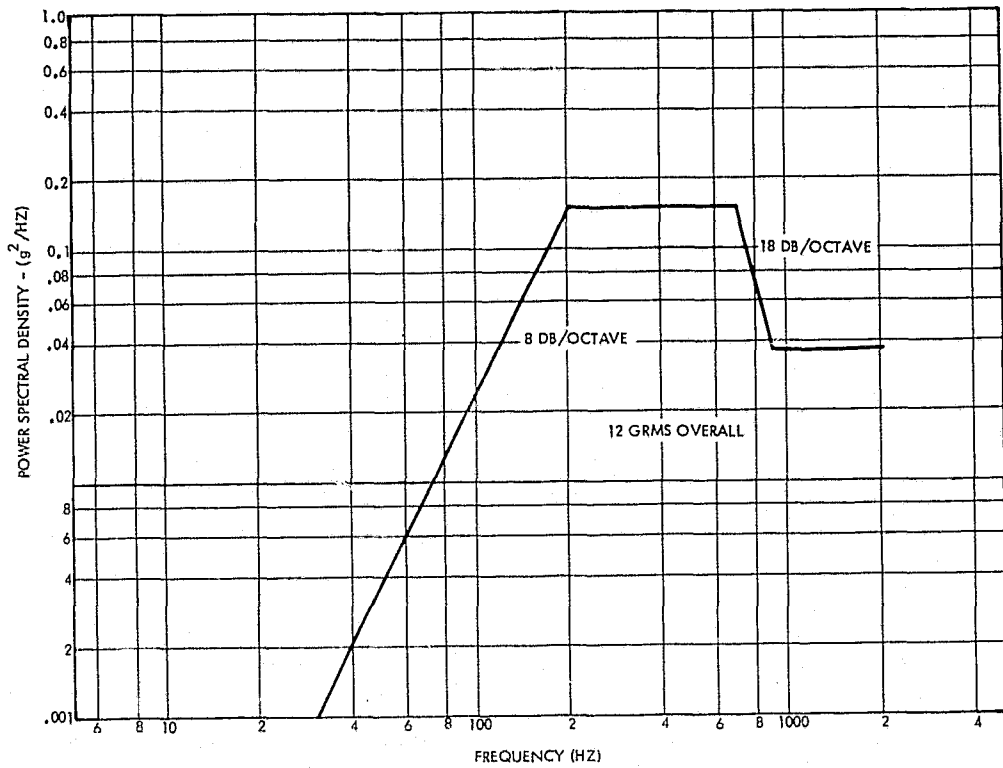


Figure 1-5. Random Vibration Spectrum Used for Dynamic Analysis and SECT Program

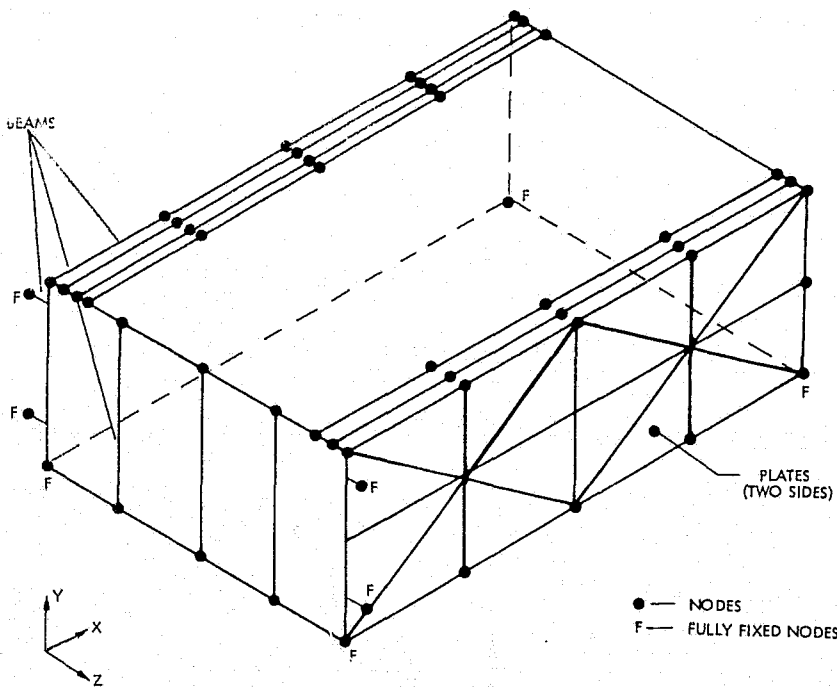


Figure 1-6. Structural Model of CAMAC Crate Used for Dynamic Analysis

The mass applied at each node is assumed to be acting simultaneously along all the orthogonal axes, which does not affect uncoupled modes. The forces applied for the static stress analysis were also applied simultaneously. However, the extreme fiber bending stresses were calculated at cross-section positions to reflect load direction independently.

The dynamic model assumes the modules are held in place by two fasteners. For commercial applications, only a single attachment screw is located at the bottom of the module front panel. For flight application, it would be necessary to include a second fixing screw at the top of the front panel similar to the NIM modules. The CAMAC standards for the crates make optional a threaded-hole pattern on the top crate rail to accept NIM modules. Most manufacturers produce their crates with the same 25 threaded-hole pattern on the top crate rail as on the bottom rail. The top screw is thus a change to only the CAMAC modules.

The crate dynamic model has 284 joints, 14 constrained joints, 10 beam section properties, 283 members, and 32 plate elements. Full fixity boundary conditions were assumed at the four corners of the bottom surface and at two points on the edges of the two front panel mounting flanges. All other surfaces, including the top, were considered free in all six directions, with the exception of plate element nodes which were rotationally fixed in the local vertical direction.

1.3.1.3 Module Dynamic Model

The module dynamic model has 15 joints, 13 constrained joints, 3 section properties, 12 beam members, and 16 plate elements. The model is depicted in Figure 1-7. The module was considered to be pinned in the crate guide rails (lateral movement prevented but no rotational constraint). All plate element nodes were considered rotationally fixed in the local vertical direction.

1.3.1.4 Analysis Criteria

Stress Criteria - Material properties will be minimum as specified in MIL-HDBK-5. A factor of safety of 1.25 with respect to the material yield is desired for stresses on structural members; plastic yielding will not be allowed.

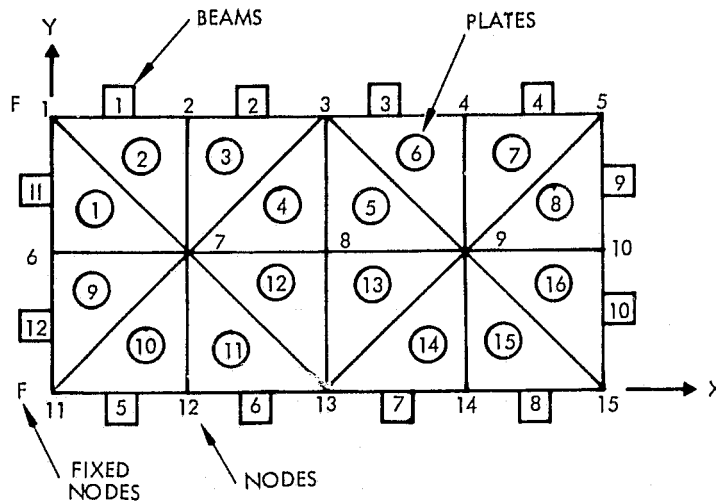


Figure 1-7. Structural Model of CAMAC Module Used for Dynamic Analysis

Stiffness Considerations - Frequently, allowable deflections rather than stress will control the design of electronic equipment structure. This is particularly true for printed circuit boards, connector mounts, and, in general, wherever wiring interconnections are of prime consideration. Maximum printed circuit board deflections should be below 0.1 inch, double amplitude to avoid problems.

Design Load Factors - The design load factor is an equivalent static acceleration by which the mass of a resonating structural member is multiplied to obtain an equivalent static load for purposes of calculating stress.

The design load factors are determined from the peak Rayleigh three-sigma acceleration, G_p , which is given as a function of frequency by:

$$G_p = 3\left(\frac{\pi}{2} Q f W\right)^{1/2},$$

where Q is the assumed transmissibility, f is the frequency, and W is the random vibration power spectral density at frequency, f . The peak acceleration calculated from the random vibration spectrum shown in Figure 1-5 with an assumed transmissibility of ten is plotted in Figure 1-8. The assumed transmissibility of ten is toward the lower end of the values expected for this type of hardware. A maximum expected value would be twenty.

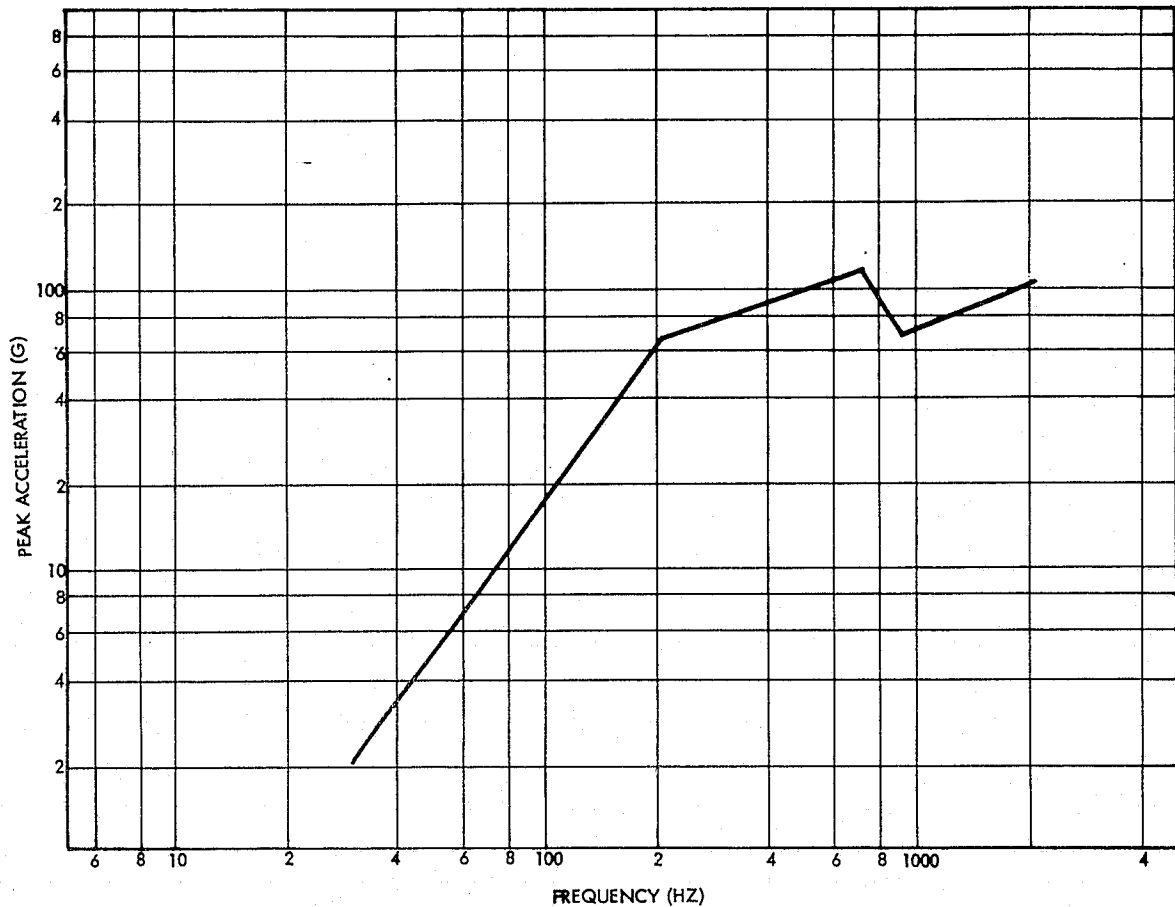


Figure 1-8. Three-Sigma Peak Acceleration Spectrum Used for Dynamic Analysis

The design load factor at a particular resonant frequency, f_n , of a structural member is the value of the peak acceleration at f_n given in Figure 1-8 after multiplication by a correction factor ranging from 0.5 to 1.0 to take into account the particular structural configuration.

1.3.1.5 Dynamic Analysis Results and Conclusions

Natural Frequencies - The calculated natural frequencies of the crate structure and the module printed circuit boards are listed in Table 1-4. The fundamental and second mode frequencies are relatively low, which will result in significant motions. Cable harnesses and wiring will, therefore, need to be supported along their lengths to avoid overstressing at cable terminations.

Table 1-4. Calculated Natural Frequencies of the
CAMAC Crate/Module System

<u>Crate Natural Frequencies</u>		<u>Module Printed Circuit Board Natural Frequencies</u>	
Mode	Frequency (Hz)	Mode	Frequency (Hz)
1	53	1	53
2	92	2	101
3	125	3	220
4	128	4	251
5	130		
6	139		

Crate Dynamic Stresses - Basic structural elements are stressed well below nominal yield stresses. The maximum stress found was 4806 psi which gives a margin of safety of 1.9, based upon a yield value of 14,000 psi. The stress levels are sufficiently low that fatigue of the basic structure will not be a consideration. However, all bolts and screws will need to be properly torqued and locked.

Module Dynamic Stresses and Deflection - The maximum dynamic stress found was 600 psi, which is well below the levels allowed in standard aerospace practice. The maximum printed circuit board deflection was 0.044 inch, double amplitude. Although significant, this value is not as large as one would expect intuitively. There is adequate clearance to preclude collision between adjacent boards. However, the deflections are sufficiently large to require attention to wire routing and attachment to the board. Stress relief and spot bonding will be required. Flexing of part leads due to board curvature is normally acceptable for board deflections up to 0.07 inch, double amplitude at a minimum. Stiffening or additional support to the boards should, therefore, not be required to preclude part lead failures.

Peak Accelerations - The peak accelerations shown in Figure 1-8 are, in general, 25 to 35 percent of the peak levels typically seen in spaceflight hardware. Since the types of parts used in NIM/CAMAC equipment are physically similar to space-qualified parts, we do not expect internal part failures to be a concern. The only components that might be susceptible to the predicted peak accelerations are electromechanical devices such as switches, circuit breakers, relays, and crystals. With the exception of switches, such devices are only rarely used in NIM/CAMAC equipment.

The overall result of the dynamic analysis is that the calculated stresses, deflections, etc. are all below the levels that are typically encountered in spaceflight equipment. Therefore, only relatively minor structural modifications should be required to make NIM/CAMAC equipment compatible with the Spacelab dynamic environments. This conclusion is further strengthened by the fact that the random vibration level actually used in the analysis is well above the most recent expected value (see Section 1.2.2).

1.3.2 Thermal Analysis

The thermal analysis addressed the use of NIM/CAMAC equipment in the forced-air, convective-cooled environment of the Spacelab module experiment racks. The primary purpose of the analysis was to determine the following:

- the forced convective heat transfer coefficient as a function of air flow rate and location on the printed circuit boards,
- the maximum board temperature as a function of air flow rate and power dissipation,
- the maximum part case temperature as a function of power density and air flow rate.

In addition, the air flow distribution system for the rack-mounted equipment was reviewed, and test techniques were recommended. The analysis is described in detail in TRW Report 7517.1-348, "Thermal Study of NIM/CAMAC Rack-Mounted Experiment Equipment for Spacelab Application," April 6, 1976.

1.3.2.1 System Configuration

The crate or bin of the NIM and CAMAC systems contains equipment modules with their electronic parts mounted on printed wiring boards, and the modules are installed vertically into slots in the housing structure. Openings are provided on the top and bottom housing structures next to the boards. For Spacelab applications, an air flow distribution plenum is located on the top of the housing (see Figure 1-9). The crate or bin is mounted inside a rack (cabinet), and is connected to the distribution duct located in the back of the rack with flexible connections (see Figure 1-10).

According to the Spacelab Payload Accommodations Handbook, cooling of the crate or bin is accomplished by suction pressure in the following manner. Each rack is connected to the avionics loop supply duct and return duct, which are located under the floor of the module housing the racks. Air enters

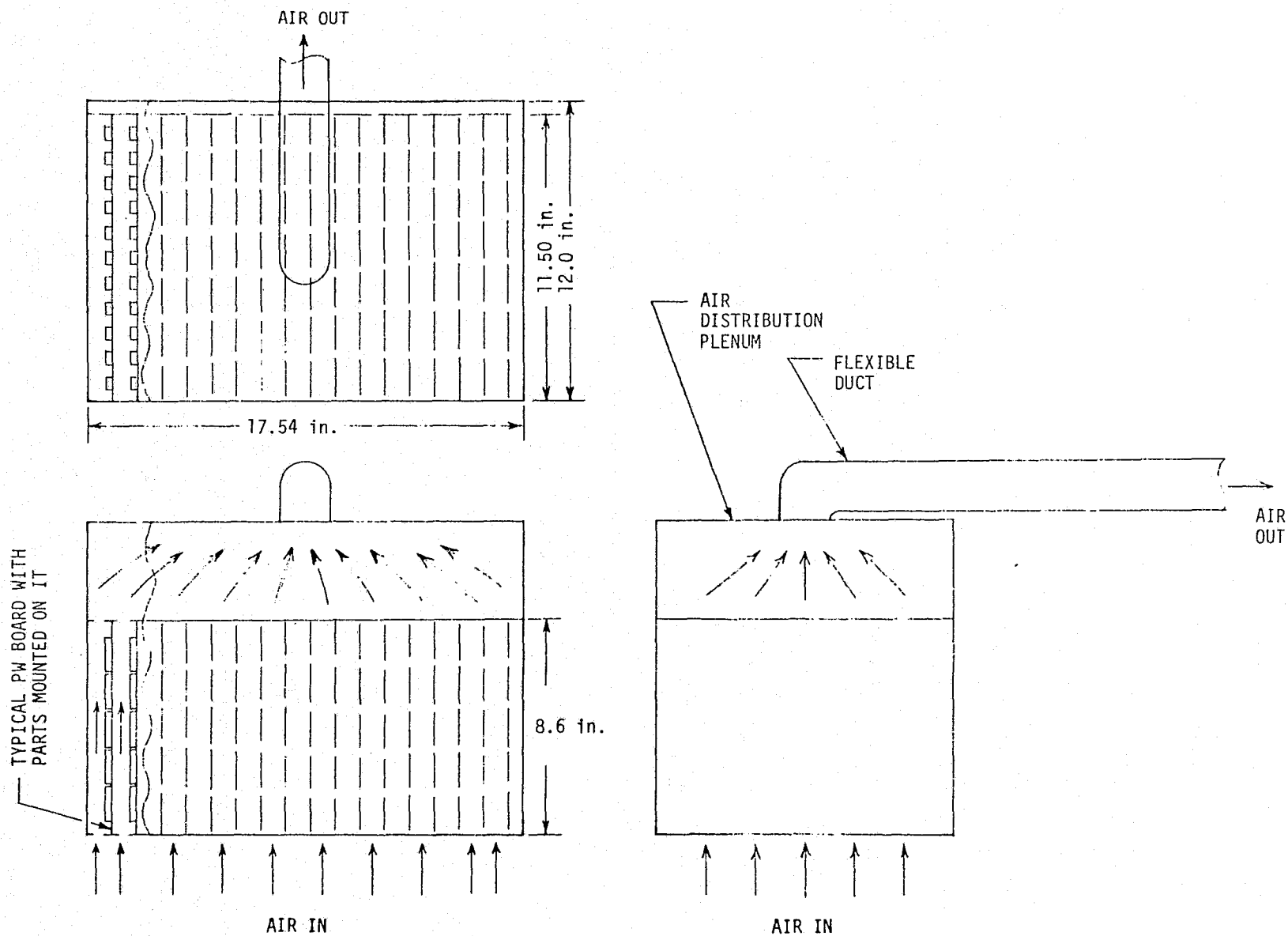


Figure 1-9. Cooling Air Flow Arrangement for Spacelab Rack-Mounted NIM/CAMAC Equipment

ORIGINAL PAGE IS
OF POOR QUALITY

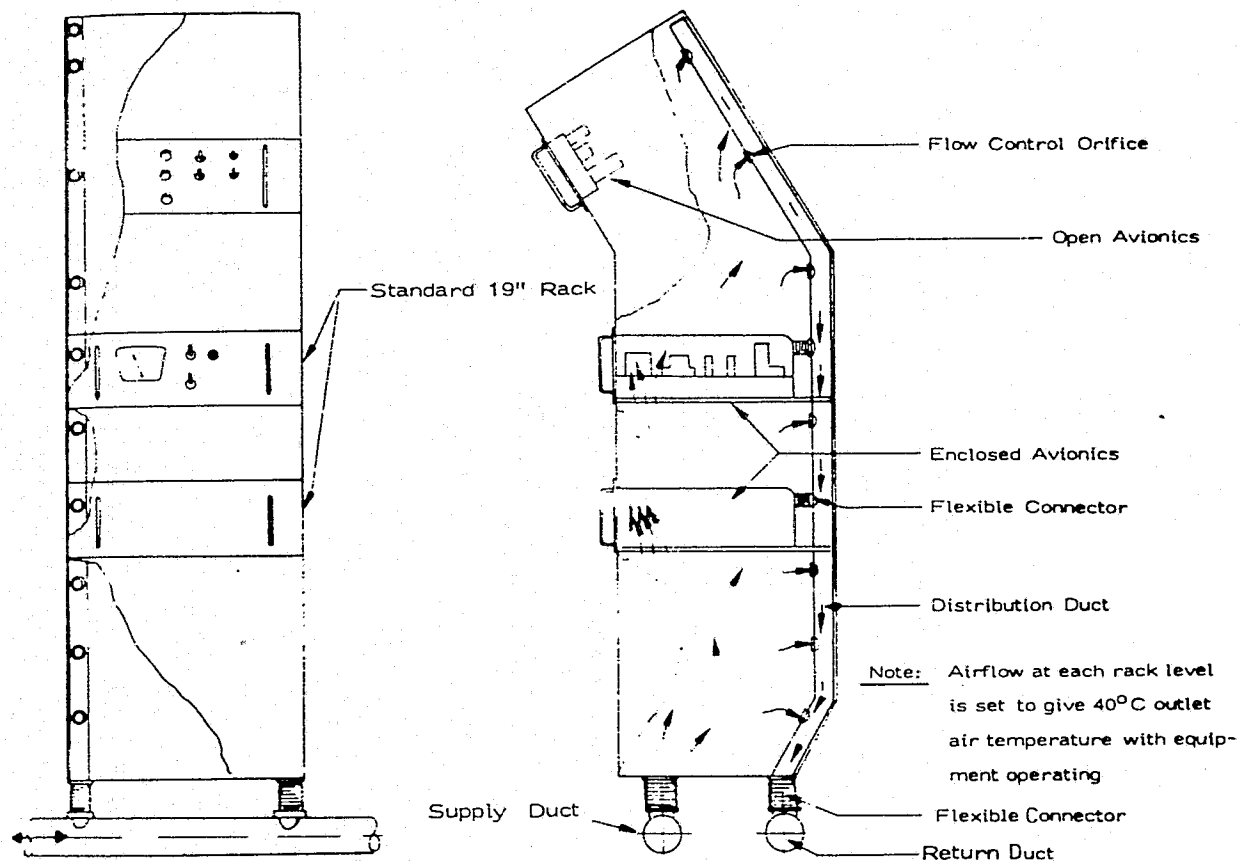


Figure 1-10. Spacelab Rack-Mounted Equipment Forced-Air Cooling System Cooling System

the rack from the supply duct, flows through the crate(s) or bin(s) from the bottom to the top, exits from the distribution plenum, and is then sucked through the distribution duct to the return duct. Each inlet to the distribution duct contains an adjustable orifice that is used to control the air flow through the connected crate or bin according to its needs. Unused inlets are capped off.

The cooling air flow distribution system described herein appears to be feasible. However, in order to achieve an equal quantity of air flow among the boards, the air flow distribution plenum must be designed properly. Detailed design of the plenum was beyond the scope of this study.

1.3.2.2 Thermal Analysis Assumptions

The basic assumptions employed in the analysis are listed below. The first two defined critical parameters of the Spacelab cooling system that had not been fixed when the analysis was started. The rest of the assumptions were made to simplify the analysis.

- Cooling air temperature is 23 °C (74 °F) inlet and 40 °C (104 °F) exit.
- Standard cooling air flow rate is 21.8 kg/hr (48 lb/hr) per 100 watts of power dissipation.
- All boards receive the same quantity of cooling air.
- All boards dissipate the same amount of power.
- The power dissipations are uniformly distributed on the boards.
- The total surface area of all parts mounted on a board equals the surface area of one side of the board.
- The convective heat transfer coefficient of the parts equals that of the board to which the parts are mounted.
- Steady-state thermal and air flow conditions prevail.
- The boards are 28 cm (11 inches) wide x 20 cm (8 inches) high.
- The effective air flow spacing between the boards is 13 mm (0.5 inch) for the crate and 25 mm (1.0 inch) for the bin.

1.3.2.3 Convective Heat Transfer Coefficients

From Figure 1-9, it is seen that the air flow spacing between the boards, formed by two adjacent boards and the front and back panels of the crate or bin, resembles a rectangular duct. The convective heat transfer coefficient, h , as a function of air flow rate and board location is shown in Figures 1-11

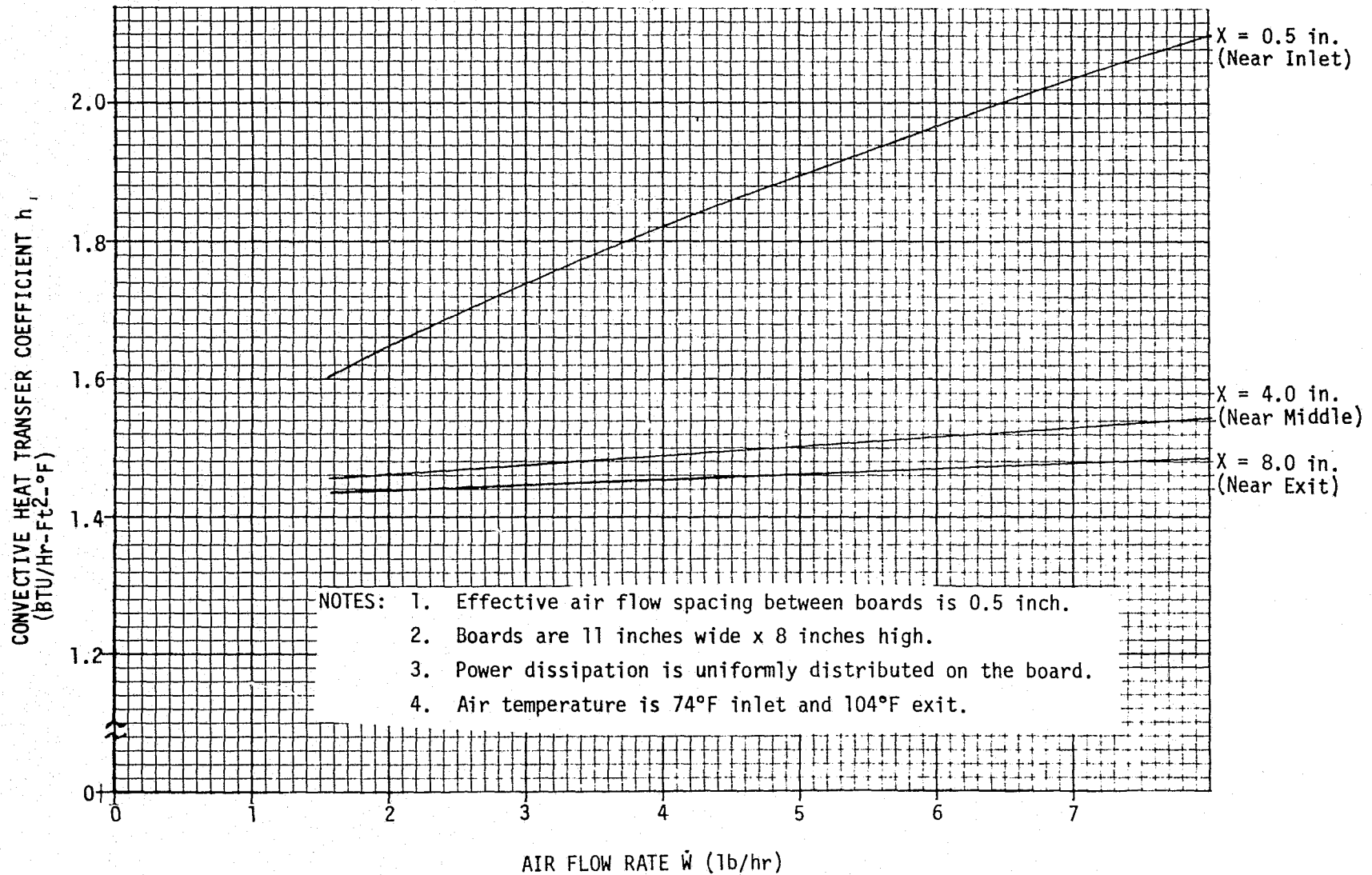


Figure 1-11. Convective Heat Transfer Coefficient as a Function of Air Flow Rate and Board Location for Crate

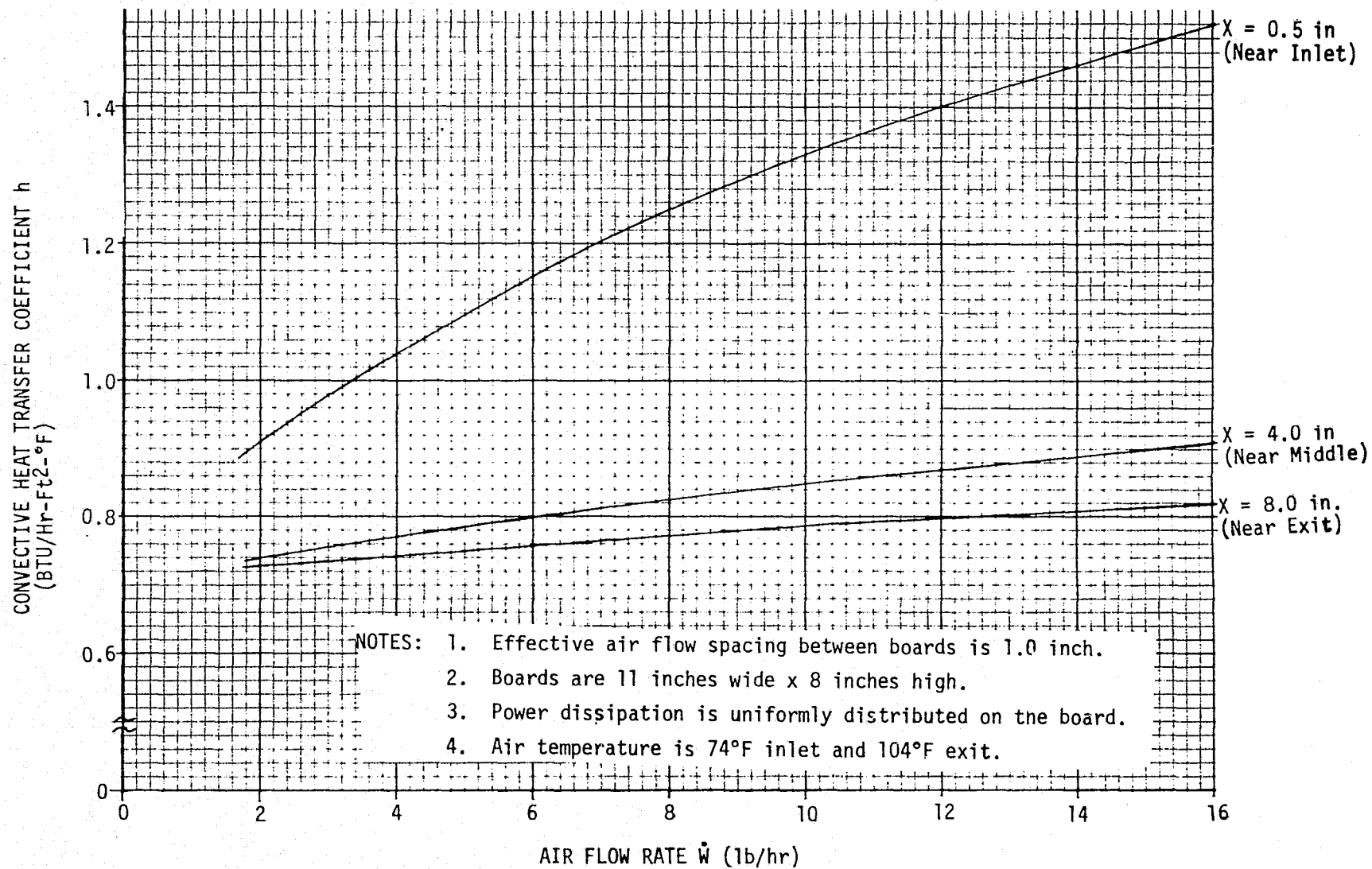


Figure 1-12. Convective Heat Transfer Coefficient as a Function of Air Flow Rate and Board Location for Bin

and 1-12. The coefficient, h , decreases rapidly with increasing flow length, X , and the difference between the h values in the middle of the board and near the exit (top of the board) is insignificant.

For a given air flow rate, power dissipation, and board location, there is a fixed temperature difference, ΔT , which consists of two components: one component, ΔT , is from the board or part case to the cooling air, which is due to heat transfer from the surface to the air. The other component, ΔT , is due to increase in the enthalpy of the cooling air, as the air absorbs the heat.

These ΔT 's change in value from inlet to exit, and the maximum value occurs near the exit where the coefficient, h , is minimum and the enthalpy is maximum. Since the worst-case maximum temperature dictates the design, the maximum ΔT 's were calculated and were added to the air inlet temperature of 74 °F to obtain the maximum board temperature.

In looking over photographs of the NIM/CAMAC equipment printed wiring boards with parts mounted on them, it was observed that most parts are DIP's (Dual In-line Packages) and the packaging density is such that the surface area of the parts approximately equals or is somewhat less than the area of one side of the board. The parts are usually separated from the board by a gap of 0.020 inch and the boards are not conformal coated. The parts are exposed to the cooling air, and the convective heat transfer coefficients of the parts are generally equal to or higher than that of the board (depending on the orientation of the parts with respect to the air flow direction). In general, the temperature difference between the case and the board tends to be insignificant. Therefore, for practical purposes, the case temperature was considered equal to the board temperature.

1.3.2.4 Thermal Analysis Results and Conclusions

The principal results of the thermal analysis are presented in the four graphs shown in Figures 1-13 to 1-16. These graphs show the relationship between maximum board or part case temperature, power dissipation and air flow rate for the CAMAC and NIM configurations. The results are presented in parameteric form to allow their use in analyzing a variety of specific cases. Our overall conclusions are based on the average or nominal situation. Several examples to illustrate the use of the results for specific cases are also given.

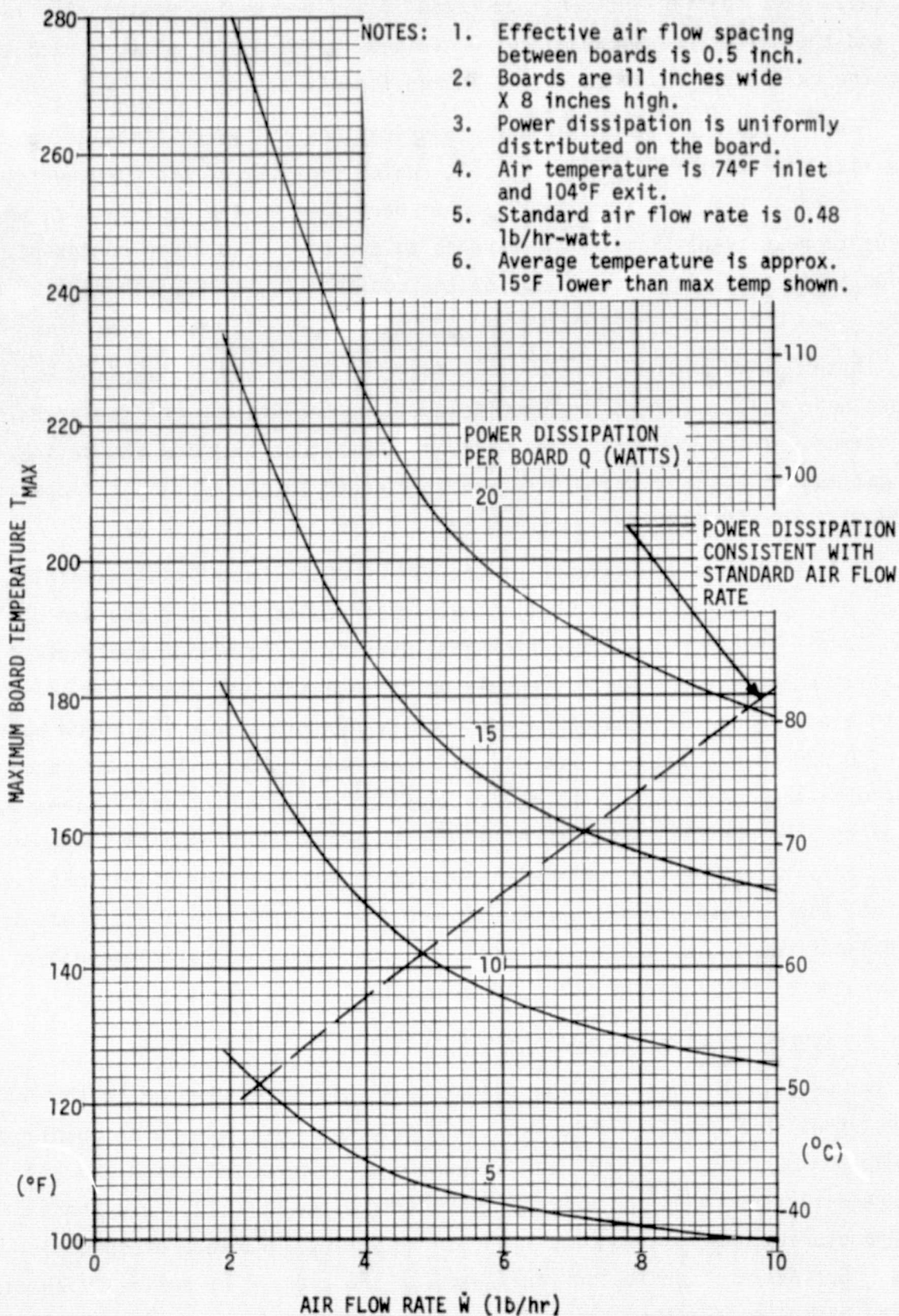


Figure 1-13. Maximum Board Temperature as a Function of Air Flow Rate and Power Dissipation for Crate

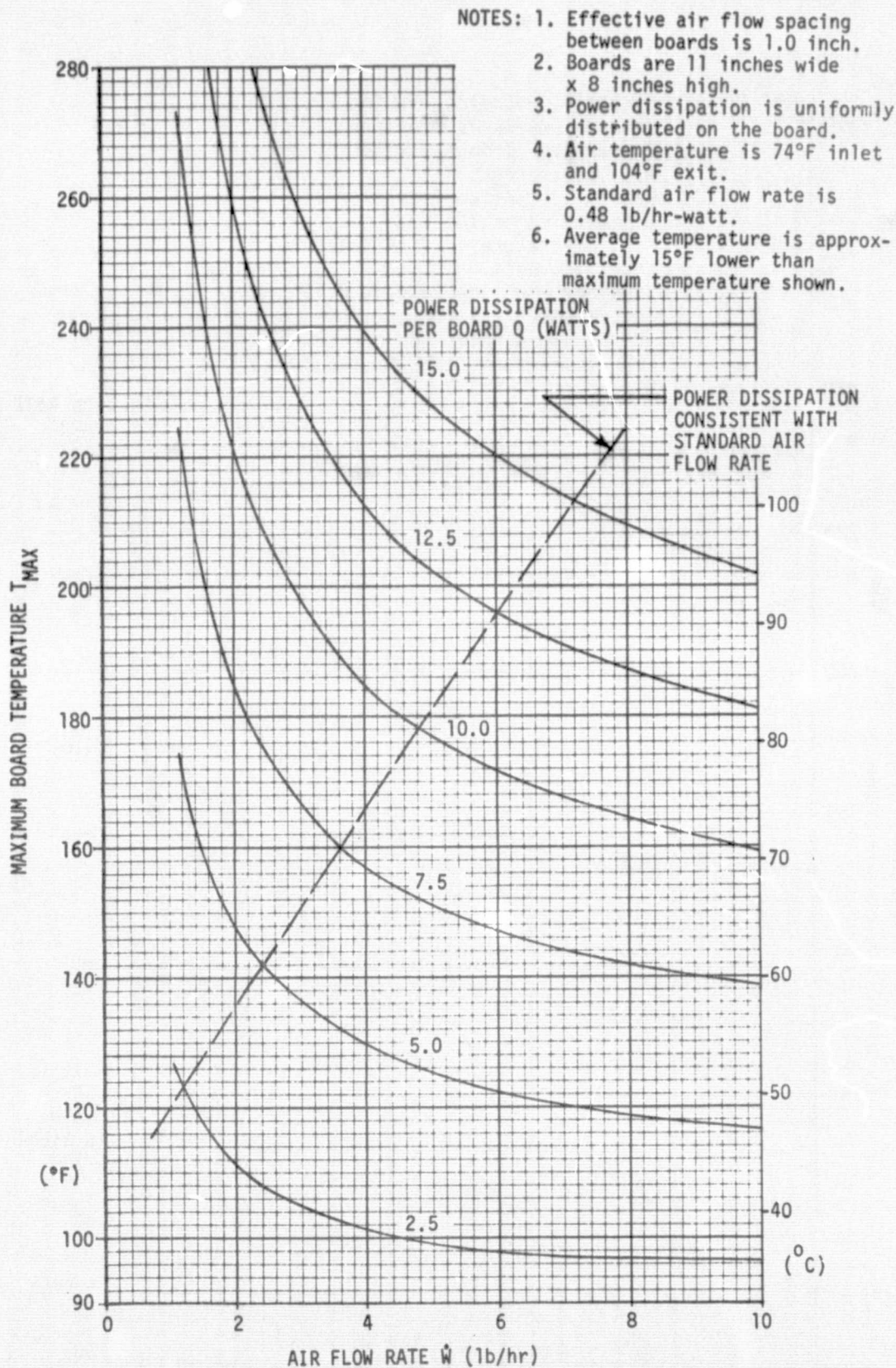


Figure 1-14. Maximum Board Temperature as a Function of Air Flow Rate and Power Dissipation for Bin

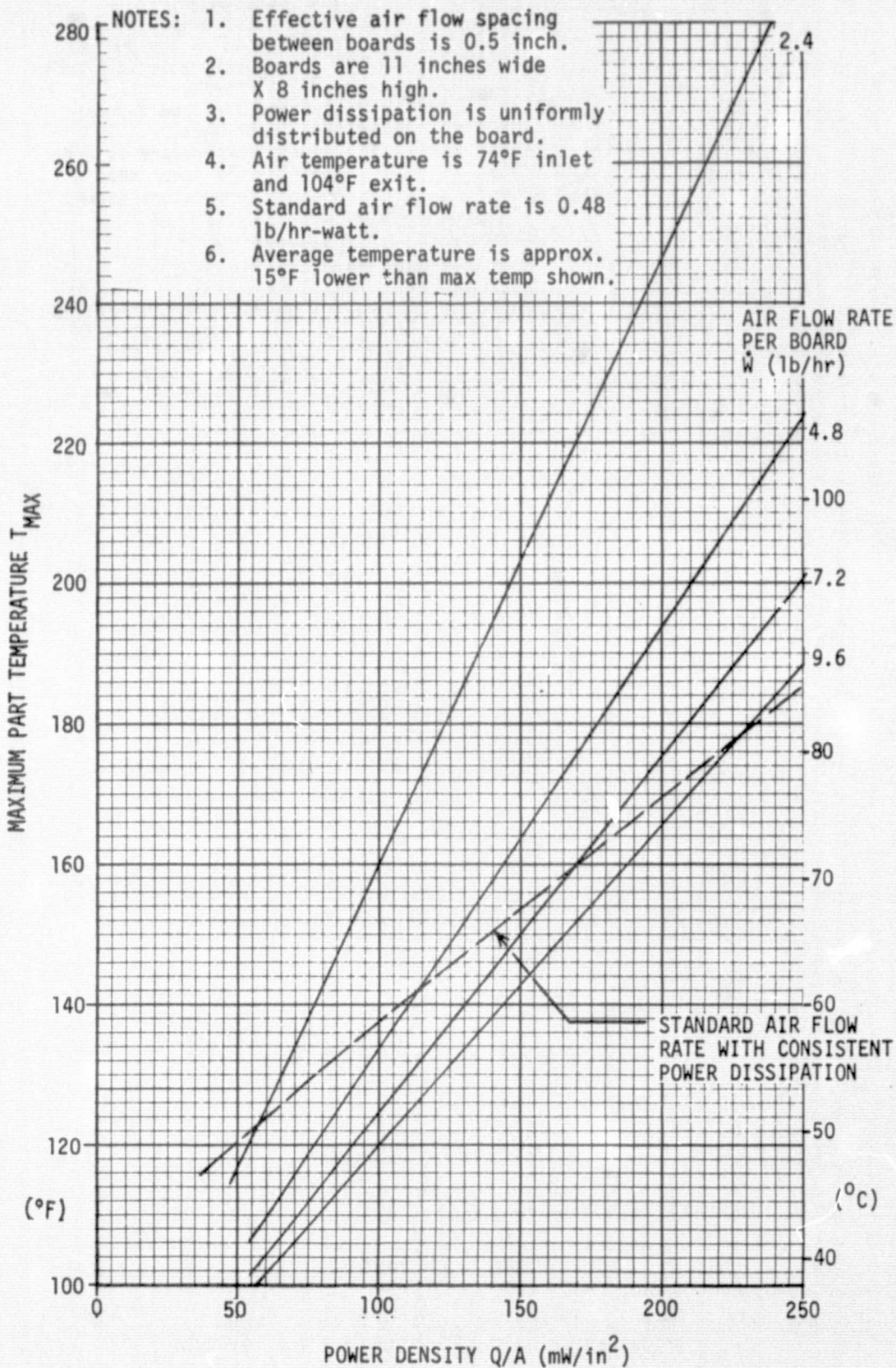


Figure 1-15. Maximum Case Temperature as a Function of Power Density and Air Flow Rate for Crate

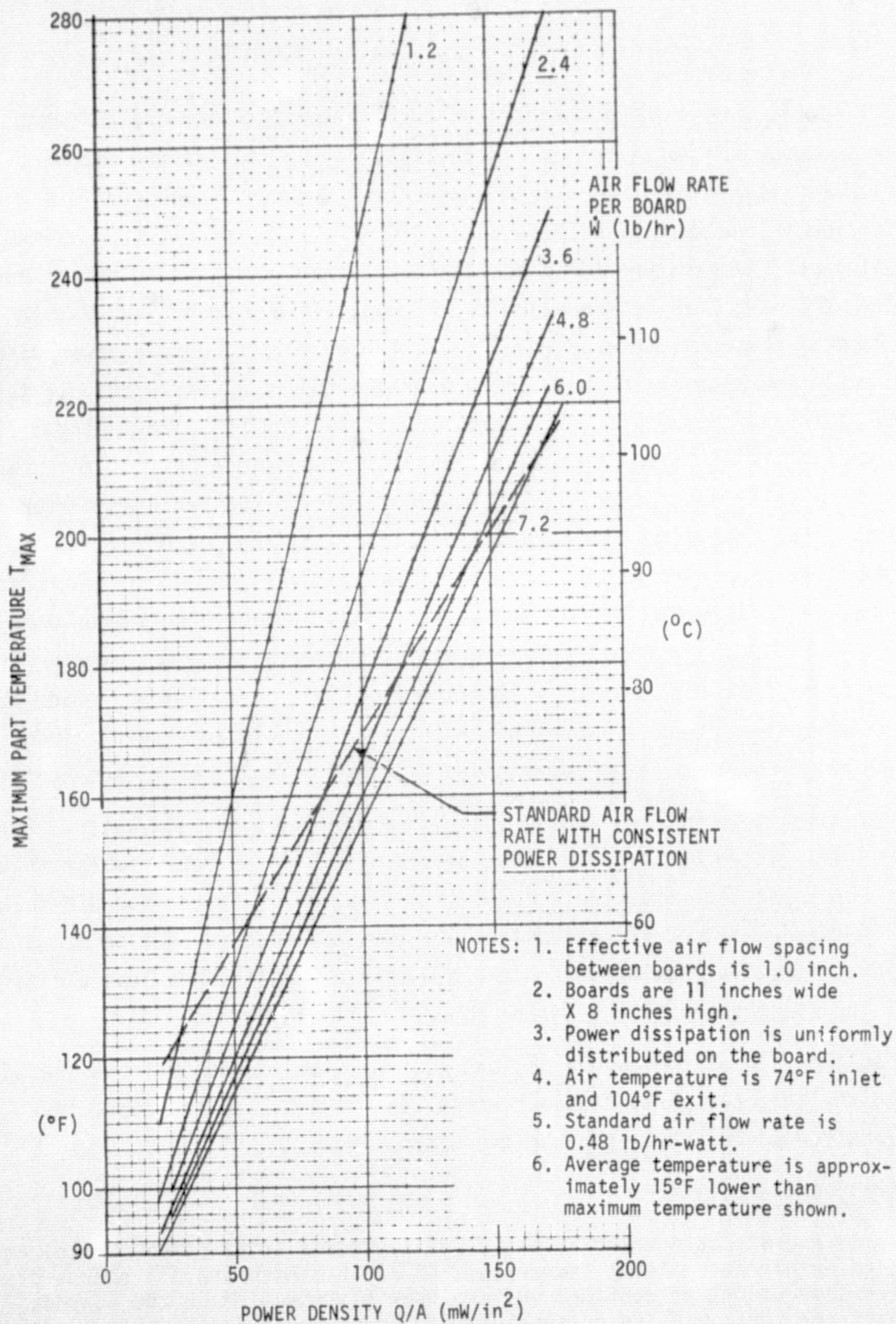


Figure 1-16. Maximum Case Temperature as a Function of Power Density and Air Flow Rate for Bin

Overall Conclusions - The average power dissipation in a commercial CAMAC module is about ten watts. From Figure 1-13, we see that at the standard SpaceLab air flow rate, the maximum board temperature, assuming uniform power dissipation in the module, will be 61 °C (142 °F). Therefore, the maximum operating case temperature for industrial grade parts (70 °C) is not exceeded. However, the margin is not very great. A twenty-five percent increase in either local power dissipation (see Figure 1-15) or total module power dissipation will increase the maximum temperature to the limit of 70 °C. If the maximum allowable temperature is increased to 125 °C (257 °F) by using military grade parts, the air flow rate can be reduced to under one-half of the standard value while still maintaining a comfortable margin on the part temperatures. Obviously, the most effective modification from a thermal standpoint would be to reduce the power dissipation by using low-power versions of the parts that are functionally acceptable. If the average module power dissipation were reduced to five watts, the flow rate could be halved and the maximum temperature of 50 °C (122 °F) would be well below the limit on industrial grade parts. Careful attention to any residual high-power parts would be needed, since they would be points of high local power dissipation and, hence, hot spots.

The average power dissipation in a NIM module is about six watts. From Figure 1-14, it can be seen that the nominal situation is very similar to the CAMAC case. The fact that the thermal situation is not better for NIM than CAMAC is due in large part to the last assumption listed in Section 1.3.2.2. The wider effective duct width for the NIM module results in a less efficient use of the cooling air flow (compare Figures 1-11 and 1-12).

Modification of the module top and bottom covers to concentrate the air flow on the internal circuit boards would tend to equalize the convective heat transfer coefficients for the two cases.

Illustrative Examples -

Example 1: A crate containing 25 boards dissipates a total of 250 watts, and is cooled by air at an inlet temperature of 74 °F. Assuming all boards dissipate the same amount of heat and uniform heat distribution on the boards, determine the maximum board temperature if the cooling air flow rate is (a) standard, (b) 50 percent of standard, and (c) 150 percent of standard.

Power dissipation per board $Q = 250 \text{ watts}/25 \text{ boards}$
 $= 10 \text{ watts/board.}$

From Figure 1-13, Note No. 5, standard air flow rate is 0.48 lb/hr-watt ,

$$\dot{W} = 0.48 \text{ lb/hr-watt} \times 10 \text{ watts} = 4.8 \text{ lb/hr.}$$

From Figure 1-13, for $Q = 10 \text{ watts/board}$,

$$\dot{W} = 4.8 \text{ lb/hr, } T_{\text{board}_{\text{max}}} = \underline{142^\circ\text{F}} \quad (\text{a})$$

$$\dot{W} = 2.4 \text{ lb/hr, } T_{\text{board}_{\text{max}}} = \underline{172^\circ\text{F}} \quad (\text{b})$$

$$\dot{W} = 7.2 \text{ lb/hr, } T_{\text{board}_{\text{max}}} = \underline{132^\circ\text{F}} . \quad (\text{c})$$

Example 2: An 11-inch wide x 8-inch high printed wiring board, with 16 lead DIP's mounted on it, is to be used in a crate. The board has a total power dissipation of ten watts, and is to be cooled with 74°F air at a rate of 4.8 lb/hr . Assuming all parts dissipate the same amount of power and have the same heat transfer coefficient, determine the part case temperature if the packaging density is such that (a) the total surface area of all parts equals the board area; (b) there are 12 rows of 13 parts each; and (c) there are 17 rows of 7 parts each.

For average cases, such as the example given herein, heat transfer from the part surface that faces the board may be assumed negligible, since there tends to be low air flow through the small gap (0.020 inch) between the part and the board. Also, heat transfer from the leads may be assumed to be negligible due to the small surface area of the leads, although the heat transfer coefficient over the leads may be high. These simplifying assumptions should yield slightly higher temperatures, which is a conservative approach. However, for extreme cases with high power dissipation, every possible heat path should be included.

$$A_{\text{parts}} = A_{\text{board}} = 11 \times 8 = 88 \text{ in}^2 \quad (\text{a})$$

$$\frac{Q}{A_{\text{parts}}} = \frac{10 \text{ W}}{88 \text{ in}^2} = 0.114 \text{ W/in}^2 = 114 \text{ mW/in}^2$$

$$\text{From Figure 1-15, } T_{\text{case}_{\text{max}}} = \underline{142^\circ\text{F}} .$$

$$A_{\text{parts}} = 0.725 \times 0.265 + 2(0.725 \times 0.160) \quad (b)$$

$$+ 2(0.265 \times 0.160) = 0.509 \text{ in}^2/\text{part}$$

$$A_{\text{parts}} = 12 \text{ rows} \times 13 \text{ parts/row} \times 0.509 \text{ in}^2/\text{part}$$

$$= 79.4 \text{ in}^2(\text{total})$$

$$\text{From Figure 1-15, } T_{\text{case}_{\text{max}}} = \underline{149 \text{ } ^\circ\text{F}} .$$

$$A_{\text{parts}} = 17 \text{ rows} \times 7 \text{ parts/row} \times 0.509 \text{ in}^2/\text{part} \quad (c)$$

$$= 61.0 \text{ in}^2(\text{total})$$

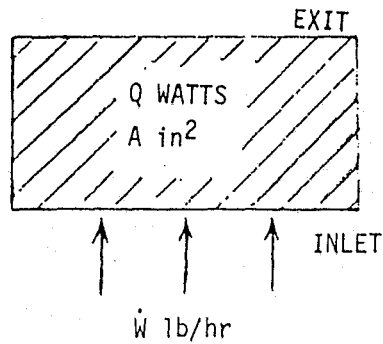
$$\frac{Q}{A_{\text{parts}}} = \frac{10 \text{ W}}{61 \text{ in}^2} = 0.164 \text{ W/in}^2 = 164 \text{ mW/in}^2$$

$$\text{From Figure 1-15, } T_{\text{case}_{\text{max}}} = \underline{173 \text{ } ^\circ\text{F}} .$$

The foregoing example shows the importance of surface area. For the same power dissipation and air flow rate, the temperature increases as area decreases, and vice versa. Also, when the total surface area of all parts equals the board area, the case temperature equals the board temperature for the same air flow rate and heat load as indicated in Examples 1(a) and 2(a). It should be alerted that, when using Figures 1-13 and 1-14, one must pay attention to the difference between the total part surface area and board area, because for a reduction in area of 31 percent (from 88 in² to 61 in²), the temperature increased 31 °F (from 142 °F to 173 °F) [see Examples 1(a) and 2(c)]. If one knows that the total part surface area is less than the board area, he should use Figures 1-15 and 1-16 instead of Figures 1-13 and 1-14 to estimate the temperature, as was done in Example 2.

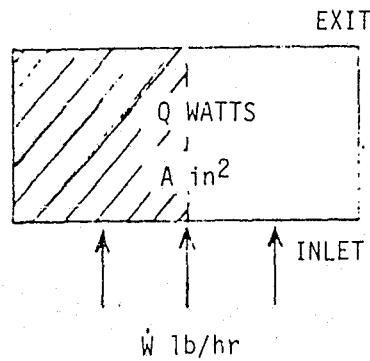
The study considered mainly the ideal case with the heat load uniformly distributed on the board. For less ideal cases, such as heat load concentrated on the left or right half of the board, or the upper or lower half of the board, the results developed for the ideal case can be used to estimate the temperatures with sufficient accuracy when simple factors are applied, as shown in Figure 1-17 and the following example:

CASE 1. Heat Load Uniformly Distributed on the Board:



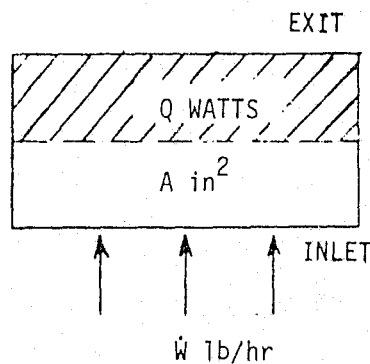
USE CURVES AS THEY ARE

CASE 2. Heat Load Concentrated on Left or Right Half of the Board:



USE CURVES FOR $Q_{\text{effective}} = 2Q$

CASE 3. Heat Load Concentrated on Upper or Lower Half of the Board:



USE CURVES FOR $Q_{\text{effective}} = 1.5Q$
AND ADD 20°F TO T_{MAX} OBTAINED

Figure 1-17. Cases for Uniformly Distributed and Concentrated Heat Load Conditions

Example 3: Same as Example 1 except instead of uniform heat distribution, heat is concentrated on the (a) left half of the board; and (b) upper half of the board. Determine the maximum board temperature for a cooling air flow rate of 4.8 lb/hr (standard).

From Figure 1-17, Case No. 2, determine the temperature by using (a) the curves for $Q_{\text{effective}} = 2Q$.

From Figure 1-13, for $Q = 2 \times 10 = 20 \text{ W}$, $T_{\text{board}_{\text{max}}} = 210 \text{ }^{\circ}\text{F}$.

From Figure 1-17, Case No. 3, determine the temperature by using (b) the curves for $Q_{\text{effective}} = 1.5 Q$ and add $20 \text{ }^{\circ}\text{F}$ to the temperature so obtained.

From Figure 1-13, for $Q = 1.5 \times 10 = 15 \text{ W}$, $T_{\text{board}_{\text{max}}} = 176 \times 20 = \underline{196 \text{ }^{\circ}\text{F}}$.

1.3.2.5 Recommendations for the SECT Program

As part of the thermal analysis effort, more definitive recommendations for thermal testing in the SECT program at JSC were generated. The objective of the tests, as described here, is to simulate the Spacelab environment for rack-mounted equipment. The recommended configuration is shown in Figure 1-18. The equipment will be operated in a laboratory ambient environment with ambient air sucked through it to provide cooling. An air distribution plenum, with a long flexible duct attached to it, will be installed on top of the crate. Located near the other end of the flexible duct is a fan, which will be utilized to suck the air through the crate.

For this test, the following measurements should be performed:

- Total electrical power input to the crate, and if possible, power input to some typical and high-power boards (same boards whose temperatures are to be measured).
- Temperatures of typical and high-power boards and parts mounted on them; at least two boards located in the middle and two boards on one side of the crate, and the high-power dissipating and low-power dissipating parts.
- Total cooling air flow rate through the crate, and preferably air flow rates over typical and high-power boards, as well as the middle and side boards (whose temperatures are to be measured).

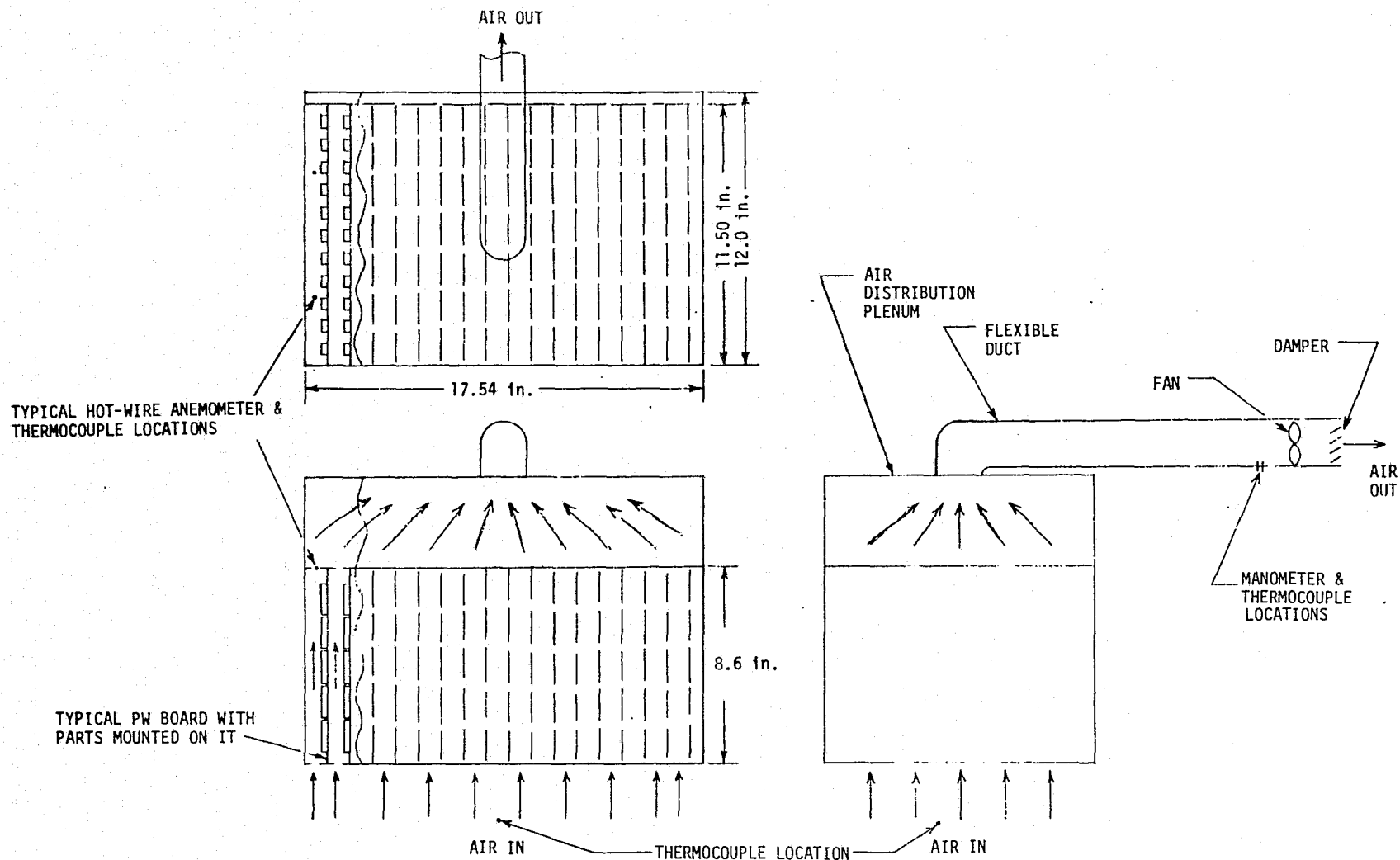


Figure 1-18. Recommended Test Techniques for Thermal Testing of NIM/CAMAC Experiment Equipment

Every attempt should be made to measure the power dissipations of the boards, because without knowing the values of this parameter, comparison of the test and analytical results would not be meaningful.

For temperature measurements, it is recommended that thermocouples be bonded to the boards and parts with a thermally conductive epoxy.

To determine the total air flow through the crate, an inclined water manometer can be used to measure the static pressure across the fan, and from the fan characteristic curve, the volume air flow rate can be determined. In order to calculate the weight air flow rate, the air temperature upstream of the fan should be measured, from which the density of air can be determined. It is desired to test the equipment at the following air flow rates: (a) standard, (b) 50 percent of standard; and (c) 150 percent of standard. The air flow rate can be varied with a damper that should be located downstream of the fan.

The adequacy of the air distribution plenum can be determined by measuring the air flow rates over the boards that are located near the middle and near one side of the crate. Calculations showed that the air velocity over the boards is approximately 30 ft/min, and in order to measure this low velocity with sufficient accuracy, it is recommended that hot-wire anemometers should be located in the exit openings (upper structure of the crate which keeps the boards in the vertical position), where the area can be measured accurately. The air temperatures at the same locations should be measured with thermocouples, which allow the determination of the density of air. Finally, the weight air flow rate can be determined by the equation of continuity.

1.4 RECOMMENDED MODIFICATIONS AND ESTIMATED COSTS (TASK 3B)

1.4.1 Recommended Modifications

The preliminary assessment of the modifications that would be required to make NIM/CAMAC equipment compatible with the Spacelab environments, presented in Section 1.2.3, was reevaluated in light of the results of our dynamic and thermal analyses to arrive at our final recommendations for equipment modifications. The modifications are treated in two categories. First, we will consider the minimum modifications that are required to use NIM/CAMAC equipment with at least some degree of confidence in its reliable operation. Second, we will consider those modifications that would essentially assure reliable operation. At the same time, these more extensive modifications will allow the incorporation of the changes needed to alleviate the convective cooling requirements of NIM/CAMAC equipment.

1.4.1.1 Module Modifications

In general, the minimum modifications required correspond to the preliminary assessment except in the area of structural changes where the situation was found to be better than our intuitive judgment indicated. The minimum module modifications required are given in Table 1-5. They apply to both NIM and CAMAC modules, with the few exceptions noted. These modifications are all of the type that could be performed on a commercial module after its original fabrication with the possible exception of the isolation of circuit and frame grounds. Changes in the printed circuit board layout, which are difficult to implement after the fact, may be required to obtain ground isolation.

The second class of modifications considered for NIM and CAMAC modules are more extensive and involve a significant amount of redesign prior to module fabrication. These modifications could not be performed on commercial modules after their original fabrication. The general approach would be the following: starting with the existing circuit design, perform design and reliability analyses directed at reducing power consumption, replacing commercial electronic components such as parts and connectors with items selected from a NASA-approved parts list, and increasing the circuit reliability under worst-case conditions. Once the circuit redesign is completed,

Table 1-5. Minimum Modifications Required for
NIM and CAMAC Modules

- Conformally coat the printed circuit boards.
- Mechanically support or spot bond unsupported parts weighing more than five grams and vulnerable parts such as vertically-mounted capacitors.
- Stress relieve and spot bond point-to-point wiring.
- Lock all fasteners by spot bonding or substituting self-locking hardware.
- Isolate circuit ground from frame ground.
- Review materials (especially plastics) and replace with acceptable materials; e.g., Teflon-insulated wire.
- Analyze power dissipation to identify local hot spots. Correct by installation of heat sinks or replacement with extended temperature range part.
- Inspect soldering and lead forming on parts. Rework or replace suspect items as required.
- Install top front panel attachment screws on all CAMAC modules.
- Install guide pins to provide mechanical support for the rear card edge connectors on all CAMAC modules.
- Review electromechanical devices such as switches, potentiometers, etc., and replace with vibration-qualified devices or hard-wired parts. These devices are mostly found in NIM modules.

a new product design, primarily involving a modified printed circuit board layout, would be performed. Again, this board layout would presumably start with the existing layout and incorporate both the circuit and component changes as well as the modifications identified as minimum modifications in Table 1-5. The fabrication of the redesigned module would be done in conformance with current NASA-approved processes and assembly techniques for space electronics. The inspection and test activities would also be handled in much the same way as they currently are for experiment electronics to be flown on unmanned scientific spacecraft.

This approach to implementing NIM and CAMAC equipment for spaceflight experiments is being actively pursued by NASA/GSFC. Three manufacturers of commercial CAMAC and NIM equipment are each investigating this type of approach for several specific modules under contract to GSFC (see Table 1-2, Volume II).

One of the objectives of the GSFC work is to reduce the power dissipation to the point at which conduction cooling of the spaceflight versions of NIM/CAMAC equipment is possible. If this is feasible, some further mechanical changes to the modules will be required to improve the heat conduction paths inside the modules and from the modules to the crate or bin. These changes would be incorporated into the new product design for the modules discussed in the preceding paragraph.

1.4.1.2 Crate or Bin Modifications

The minimum modifications that must be made to the CAMAC crate or NIM bin correspond very closely to our preliminary assessment. The essential result of the dynamic analysis was to confirm that the basic structure was adequate with a comfortable margin of safety. The minimum crate/bin modifications required are listed in Table 1-6.

Table 1-6. Minimum Modifications Required for
NIM Bins and CAMAC Crates

- Provide bottom surface mechanical support and attachment to Spacelab rack structure.
- Provide top surface plenum with connection to the Spacelab cooling air return ducts.
- Lock all fasteners by spot bonding or substituting self-locking hardware.
- Add retaining mechanisms to all card guide rails.
- Add attachment points for module top front panel screws (already on NIM bins and some CAMAC crates).
- Provide guide pin sockets for CAMAC dataway connectors.

As was discussed in Section 3 of Volume II, because of the inherent high degree of commonality present in the requirements for system-common equipment, such as the crates or bins, a reasonable effort can justifiably be invested in developing spaceflight versions of the equipment. This is not particularly significant so far as the crate mechanical structure is concerned because the existing design is basically adequate. However, this point has a very significant influence on the choice of the best approach for the crate or bin low-voltage power supply. The most reasonable approach

is to design, develop, and qualify a crate low-voltage power supply specifically for Spacelab use. The redesign would be primarily directed at increasing the supply efficiency and reducing the supply weight. If a sufficient weight reduction is achieved, the supply could be mounted on the rear of the crate or bin as it is in the existing equipment. Otherwise the supply could be independently mounted to eliminate the large cantilevered load on the rear of the crate or bin.

To a large extent, this approach corresponds to going directly to the second category of more extensive type modifications discussed in the previous section for modules. Again, if it turns out to be feasible to reduce the power consumption to the level at which conductive cooling can be considered, more extensive mechanical modifications to the crate will be needed to improve the thermal conduction paths.

1.4.2 Modification Costs

In estimating the modification costs, we have taken advantage of the inherent commonality of NIM and CAMAC equipment. The types of modifications that have to be performed are relatively independent of the particular function or supplier of the module. Therefore, we have estimated the modification cost for an average single-width module (i.e., containing one printed circuit board). As a point of reference, the current average retail price of NIM or CAMAC modules is about \$700.

The actual modification cost for any particular module may vary considerably from the average cost we have estimated depending on the complexity of the unit and the amount of modification needed (see, for example, the results of the GSFC-sponsored studies by three CAMAC manufacturers). However, the principal use of our cost estimates will be to generate programmatic cost estimates in Task 4. As will be seen in the discussions of Task 4, the number of modules involved in the programmatic estimates is large. Thus, to well within the overall accuracy of the programmatic cost estimates, any variations in actual modification costs for different types of modules will average out.

Cost estimates were developed for three cases. The first two correspond to the minimum modification approach and the more extensive modification approach discussed in the previous section. The third case deals with a

so-called custom-built approach that corresponds closely to current aerospace practice for the development of unmanned spaceflight experiment electronics. This case is included for the purpose of comparing the cost of using modified NIM/CAMAC equipment for Spacelab payloads with the cost of continuing to use current methods of implementing payload electronics.

Since the crates and bins, with their associated power supplies, constitute a small fraction of the projected equipment usage (less than seven percent), no separate cost estimate was generated. As discussed in Section 1.4.1.2, the best approach for these equipment items is a one-time development of a version specifically designed for spaceflight use. Although the powered crate or bin is a distinctly different type of hardware compared with the modules, the development and production costs are not expected to be sufficiently different from the module costs to significantly affect the overall programmatic cost estimates.

1.4.2.1 Minimum Modification Costs

The minimum modifications needed to adapt NIM/CAMAC modules for Spacelab use (see Table 1-5) are essentially identical to the type of modifications identified in the Rockwell analysis of commercial equipment for Spacelab payloads. Therefore, the Rockwell cost estimates have been used as the basis for our minimum modification cost estimates. The basic approach investigated by Rockwell involved modification of an actual commercial unit after its original manufacture. As stated in Section 1.4.1.1, this approach is applicable for what we call minimum modifications.

In the Rockwell study, a wide variety of commercial equipment that was likely to be used in Spacelab payloads was analyzed. Four items of NIM equipment were included in the analysis. Two of the items, the NIM bin power supply and the Nuclear Data multichannel analyzer, are not good representations of NIM/CAMAC modules. Although the multichannel analyzer contained three NIM modules, the majority of the hardware was not NIM and the module modification costs were not separately identified. In the case of the two other items, the Tennelec timer consisted of three NIM modules and the ORTEC particle counter consisted of four NIM modules.

A certain amount of analysis and interpretation was necessary to extract module modification costs from the Rockwell report in a suitable form for our

purposes here. Most importantly, the nonrecurring engineering and test costs estimated by Rockwell for the items composed of several NIM modules took into account the fact that the individual modules are very similar. Our interpretation of their cost estimates for the Tennelec timer and the Ortec particle counter is that the engineering and test costs are independent of the number of individual modules in the equipment. It is generally true of the Rockwell modification cost estimates that the engineering and test costs are relatively insensitive to the retail price and complexity of the equipment. Therefore, we have taken the mean of the engineering and test costs for the Tennelec timer and the Ortec particle counter as the best estimate for a single module. It is certainly true that in actuality advantage would be taken of the commonality of NIM/CAMAC equipment to reduce the modification costs. However, for consistency, all of our cost estimates are based on the assumption that each module is treated independently since we cannot see a nonarbitrary way of choosing, at this point, how many modules would be modified by one organization.

The best estimate of the average manufacturing cost per module is straight forward. The total manufacturing cost for the Tennelec timer and the Ortec particle counter was divided by seven, the number of modules involved. For documentation and program management, Rockwell ratios of ten percent and five percent, respectively, of engineering, manufacturing and test were used. Finally, the costs were divided into nonrecurring design, development, test and engineering, and recurring unit costs. The Rockwell test category includes the costs associated with development verification testing of the first unit which is basically nonrecurring whereas the calibration and testing of subsequent units is included in the manufacturing costs.

The resulting cost estimate is shown in Table 1-7. A check on our interpretation of the Rockwell cost estimates can be obtained by comparing the ratio of the modification cost to the retail unit cost in Table 1-7 with the summary plot of this quantity versus retail unit cost for NIM equipment shown in Figure 3-33 of Volume II of the Rockwell report ($\$11,512/695 = 16.5$ from Table 1-7 compared with 16 for a retail unit cost of \$700 in Figure 3-33).

ORIGINAL PAGE IS
OF POOR QUALITY

Table 1-7. Estimated Costs of Minimum Modifications
for an Average NIM/CAMAC Module
(adapted from Rockwell study)

	Nonrecurring	Recurring
Engineering	5,850	--
Verification Test	3,680	--
Manufacturing and Test	-	485
Documentation	953	49
Program Management	477	24
Original Module Cost	-	695
Totals	\$ 10,960	\$ 1,253

1.4.2.2 More Extensive Modification Costs

Our estimated costs for the more extensive approach to modification for Spacelab use, described in Section 1.4.1.1, were derived from a recent TRW study of low-cost approaches to scientific experiment implementation for Shuttle-launched and serviced spacecraft (Contract NAS w-2717). As part of that study, a cost was computed for producing standard electronic modules that were similar to NIM and CAMAC modules in function and complexity, but designed specifically for spaceflight use. The costs were generated using detailed cost estimating relationships based on our experience with developing spaceflight scientific experiment electronics. The costs are broken out in accordance with a rather detailed work breakdown structure.

The cost estimate for the modification approach under consideration here was developed by modifying the level of effort devoted to each of the tasks in the work breakdown structure to correspond to the modification approach discussed in Section 1.4.1.1. The important changes to the "low-cost approaches study" estimate for development of a standardized electronic module were:

- Start from existing circuit design and printed circuit board layouts.
- Reduce number of qualification units produced and extent of the qualification test program.
- Concentrate reliability effort into parts selection and application review, worst-case analysis, and test requirements.

The resulting cost estimate is shown in Table 1-8, broken down into major task categories as well as nonrecurring and recurring costs.

Table 1-8. Estimated Costs for More Extensive Modifications to an Average NIM/CAMAC Module

	Nonrecurring	Recurring
Design Engineering	10,750	-
Product Engineering	7,500	-
Reliability	2,500	-
Parts, Materials and Processes	4,000	2,900
Quality Assurance	3,750	400
Manufacturing	13,630	1,150
Test	7,000	380
	<u>\$ 49,130</u>	<u>\$ 4,830</u>

1.4.2.3 Custom-Built Module Costs

This case does not actually correspond to modification of an existing NIM/CAMAC module, but rather represents the costs required to develop and produce an equivalent unit using a conventional, present-day approach for experiment electronic hardware intended for use on an unmanned scientific satellite. The unit is equivalent in the sense that it satisfies the same functional requirements as the NIM/CAMAC module. Our cost estimate for this case is taken directly from the TRW low-cost approaches study discussed in the previous section. The module cost estimate generated in that study is applicable without any adjustment.

The cost estimate, broken down into major task categories and nonrecurring and recurring costs, is given in Table 1-9.

Table 1-9. Estimated Costs for a Custom-Built Equivalent of an Average NIM/CAMAC Module

	Nonrecurring	Recurring
Design Engineering	27,500	-
Product Engineering	12,500	-
Reliability	6,250	-
Parts, Materials and Processes	23,375	3,150
Quality Assurance	12,500	590
Manufacturing	34,125	2,130
Test	8,750	380
Totals	\$ 125,000	\$ 6,250

1.4.3 Analysis of Alternative Modified Equipment Sources

In their analysis of commercial equipment for Spacelab payloads, Rockwell reached the conclusion that the original manufacturer is clearly the preferred modifier. In the particular case of NIM/CAMAC equipment being considered here, we came to essentially the same conclusion for many of the same reasons. However, the choice is not as clear. For NIM/CAMAC equipment, some of the factors to be considered in determining the most cost-effective source differ significantly from the general case analyzed by Rockwell.

The extremely nonuniform procurement profile used by Rockwell (large peaks every five years) almost immediately ruled out any centralized agency because of the inefficient utilization of personnel and facilities. As we will see in the discussion of Task 4A to follow, the procurement profile for NIM/CAMAC equipment is reasonably constant. In addition, if an equipment pool is adopted, a central agency that has other related functions to perform already exists.

Secondly, the inherent commonality of NIM/CAMAC equipment due to its standardized nature reduces the advantage the original manufacturer has because of his familiarity with his own equipment.

Finally, a centralized source could take greater advantage of the commonality in the kinds of modifications that are required to reduce the noncurring costs involved. Recall that in the discussion of modification costs

in Section 1.4.2.1, it was pointed out that, whereas our cost estimate assumed independent efforts for each type of module, in actuality, nonrecurring costs should be lower when more than one type of module was being modified by the same organization. Obviously, the greater the number of modules being handled in common, the greater the cost reduction.

On the other hand, due to the many types of NIM/CAMAC modules being used and the fact that most of the manufacturers each produce all or most of the types, many more than one type of module would usually be supplied by a particular manufacturer in any case.

In addition, the original manufacturer still has many advantages in terms of familiarity and available facilities and stock. This is especially true for the case of minimum modifications. Most importantly, he can incorporate the modifications during the original manufacturing cycle, as opposed to after the fact, and, hence, avoid the costs of disassembly, discarded components, retesting, etc. For the case of minimum modifications, our conclusion is that incorporation of these modifications during the original manufacture of the equipment is the most cost-effective approach.

For the case of the more extensive modifications, the modifications clearly must be incorporated during the unit fabrication and assembly. The only question is whether the most cost-effective manufacturer is the manufacturer of the commercial unit on which the redesign was based or someone else. The current U. S. suppliers of NIM and CAMAC do not produce military or aerospace equipment (the same is not true of the European suppliers). Therefore, they have the disadvantage of not having the special facilities for, or experience with, the production of aerospace equipment to military or NASA standards.

In this case, their willingness to learn the practices, procedures, and techniques required becomes a key factor. The willingness of at least some of the NIM/CAMAC suppliers to do so has been demonstrated by the GSFC-sponsored activities. Assuming this interest and willingness continues, they may well be the most cost-effective source. On the other hand, the alternative source of a contractor with experience in producing aerospace electronics, possibly gaining access to the original supplier's familiarity with the equipment through a licensing agreement, would probably not be significantly less cost-effective.

2. IMPLEMENTATION ANALYSIS AND MANAGEMENT PLAN DEVELOPMENT FOR NIM/CAMAC EQUIPMENT (TASK 4A)

2.1 INTRODUCTION

The study effort to this point has demonstrated the general applicability of NIM and CAMAC equipment for Spacelab payloads and has defined the necessary modifications and costs involved to make NIM/CAMAC equipment compatible with the Spacelab environment. In this task, we used these results to estimate the projected usage and costs of NIM/CAMAC equipment in Spacelab payload operations during the time period of 1980 to 1991.

Two alternative general approaches to providing the NIM/CAMAC equipment required for Spacelab payloads were considered.

- A shared-equipment implementation in which the various Spacelab users draw their required complement of standard NIM/CAMAC equipment for a given flight from a common equipment pool.
- A dedicated-equipment implementation in which each of the users is responsible for procuring either their own NIM/CAMAC equipment or its custom-built equivalent.

The obvious objective of the shared-equipment approach is to take advantage of the commonality found in the NIM/CAMAC requirements of the various payloads in order to minimize the total amount of equipment, and hence cost, needed to support the Spacelab payload operations. The basic assumption, which makes this approach attractive to consider, is that by committing the NIM/CAMAC equipment needed in a particular payload to that payload for only the length of time it is actually required for a given flight, the overall efficiency of equipment utilization will be significantly increased.

Experience with NIM/CAMAC use in ground-based laboratories strongly indicates that an equipment pool is a cost-effective method of satisfying user requirements for standard modules in a situation that has many factors in common with Spacelab payload operations. A simplified version of the situation for Spacelab payloads has already been considered in the commonality analysis of Task 2 (see Section 3.9.2, Volume II). Comparison of the extreme cases; namely, completely shared equipment usage in a serial flight series of the eleven representative payloads versus completely dedicated usage in a parallel flight series, indicated that equipment sharing between payloads

would reduce the total amount of equipment required from 687 items to 217 items, in the case of CAMAC equipment; and from 406 items to 245 items, in the case of NIM equipment. Thus, an overall reduction by a factor of about 2.4 was obtained in this overly simplistic scenario.

Our work in Task 4A was directed at making a more accurate and realistic assessment of the impact of equipment sharing in the Spacelab era. Hence, we have concentrated on determining the most efficient type of equipment pool implementation for Spacelab payloads. However, in order to provide a baseline for comparison, we have also estimated the costs of dedicated equipment implementation approaches that do not involve a pool. Although a number of simplifying assumptions were necessarily made in our analysis, we believe that the overall results present a reasonably accurate projection of the cost reductions that can be achieved through the use of NIM/CAMAC in a pooled-equipment implementation approach. In actuality, many detailed factors will vary from the model used here, but the overall results should remain valid. Our approach to Task 4A was subdivided into the following four tasks:

Task 4A.1 - Develop time-phased NIM/CAMAC equipment requirements.

- Define a baseline Spacelab traffic model.
- Establish a schedule for module use by an individual payload.
- Project overall NIM/CAMAC usage in the baseline payload model using the results of Task 2 for representative payloads.

Task 4A.2 - Perform a tradeoff analysis of NIM/CAMAC equipment pool concepts.

- Define alternative pool concepts.
- Tabulate pool size requirements for the alternatives.
- Select the optimum pool concept.

Task 4A.3 - Prepare a management plan based on the recommended pool concept.

- Refine pool size requirements taking into account equipment replacement rates.
- Prepare a budgetary cost estimate for pool equipment.
- Develop a recommended NIM/CAMAC equipment procurement plan and a recommended pool management plan.

Task 4A.4 - Prepare comparative equipment cost estimates for program implementations that do not involve a pool approach.

- Tabulate equipment requirements based on no equipment sharing.
- Prepare cost estimates based on nonshared NIM/CAMAC equipment and on nonshared custom equipment.

The first three tasks will result in a management plan for the recommended pool concept. The fourth task will provide the information necessary to determine the expected cost savings from implementing an equipment pools using NIM/CAMAC equipment.

The documentation used in the performance of Task 4A is listed in Table 2-1. These references were primarily used in our effort to define a baseline Spacelab payload flight traffic model that constitutes a reasonable representation of the number of Spacelab flights that will occur between 1980 and 1991. Because Shuttle mission planning activities are still in the formative stages, the payload flight traffic model will undoubtedly change. The overall results of our analysis, for the most part, can simply be scaled with the total number of Spacelab flights in the model, unless the payload mix significantly varies.

Finally, some definitions of the nomenclature being used here should be noted. As previously discussed in Section 2.1 of Volume II, we are using the term "payload" to mean a collection of instrumentation that requires approximately the full resources available in a given Spacelab flight. A flight simply means one sequence of the operations (payload integration, launch, orbital operations, return, etc.) necessary to carry out a mission with a given payload.

Since it is anticipated that many payloads will be flown more than once, with refurbishment and possibly modification between flights, the number of flights will always be greater than the number of payloads. For the case of equipment sharing, the NIM/CAMAC equipment requirements are determined primarily by the number of flights since the equipment is not uniquely identified with a given payload. The overall equipment requirements are relatively insensitive to variations in the makeup of the payload for any particular flight as long as the overall distribution of missions among the different disciplines is not changed.

On the other hand, the equipment requirements are primarily determined by the number of payloads in the case of dedicated use, and the results are relatively sensitive to questions such as how many times is a particular payload flown without making modifications that result in new requirements for NIM/CAMAC equipment. As we will see in Section 2.5, this makes the realistic estimation of equipment requirements more difficult in the dedicated equipment case.

Table 2-1. References Used in NIM/CAMAC Management Plan Development

Summarized NASA Payload Descriptions - Sortie Payloads (SSPDA), NASA/MSFC, July 1974.

The 1973 NASA Payload Model - Space Opportunities 1973-1991, NASA/Headquarters, 1973.

Updated Flight Model for Use in Shuttle, Spacelab, and IUS/Tug Procurement and Operations Analysis (Yardley Memorandum), NASA/Headquarters, October 1974.

Shuttle Era Mission Model - Sortie Payload Missions/Experiments, NASA/GSFC, December 1974.

Spacelab Briefing for AMPS, NASA/MSFC, November 1975.

2.2 TIME-PHASED NIM/CAMAC EQUIPMENT REQUIREMENTS

2.2.1 Baseline Spacelab Flight Traffic Model

In Task 2 of this study representative Spacelab payloads for a wide range of disciplines were analyzed. In order to transform this data into a measure of the total requirements for NIM/CAMAC equipment, the Shuttle flight frequency for each of the disciplines is required. The selection of this flight traffic model is important since the subsequent definition of an appropriate pool concept, management plan and program cost is based on this model. However, the precise details of the flight traffic model are not critical, as the aim of this task is to define the order of magnitude of a NIM/CAMAC pool. After consideration of several models, the Shuttle traffic model of October 1974 was chosen as a baseline because it is in good agreement with a model generated directly from the SSPDA documents. The 1974 traffic model stipulates that there are 226 Spacelab operations out of a total of 572 Shuttle flights between 1980 and 1991.

The SSPDA documents were used to identify the experiments that are projected to be flown during each year between 1980 and 1991. An estimate of the number of flights per year that are required by each discipline was obtained by adding together the reentry weights for each of the experiments and then dividing the total reentry weight by the Shuttle experiment payload capability. The payload capability varied for each discipline as it was based on whether the discipline required the pallet-only, pallet-module or module-only mode. The numbers of flights that were required each year by the different disciplines on the basis of the SSPDA documents are tabulated in Table 2-2 along with the same information from the 1974 flight traffic model. The disciplines have been grouped in a way that corresponds to the applicability of the results for the representative payloads that were selected and analyzed in Tasks 1 and 2 of this study (see Sections 2 and 3 of Volume II). As might be expected in the early years, up through 1982, the SSPDA model indicated that substantially more flights were required than projected by the flight traffic model. However, after 1982, the agreement between the two models was good. In other words, in the early years the SSPDA document oversubscribes the number of flights whereas the 1974 flight traffic model indicates the actual maximum number of flights that are possible.

2.2.2 Payload Equipment Usage Schedule

The NIM/CAMAC equipment requirements for each discipline were based on the representative payload equipment requirements that were compiled in Task 2 of this study. The equipment requirements for each year could be obtained on a qualitative basis by simply multiplying the number of units of CAMAC and NIM that were required by the representative payloads, by the number of flights per year projected by the baseline model. However, equipment requirements generated in this manner do not take into account the length of time the equipment has to be committed to a payload for a particular flight. If the payload equipment has to be committed only for a length of time that is less than the time between flights, the actual equipment requirements would be less than the simple estimate would indicate. Conversely, if the period of commitment is longer than the time between flights, the actual requirements would be increased. In order to take this factor into account, we will examine the payload development sequence before projecting the equipment requirements.

Table 2-2. Baseline Spacelab Flight Traffic Model (Flights/Year)

	Year:	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	Totals
Astronomy														
SSPDA Model		1½	2	2	4	5½	6½	7½	3½	5	3	4	2½	47
1974 Traffic Model		-	1	2	3	2	3	4	4	4	4	4	4	35
High-Energy Astrophysics														
SSPDA Model		1½	1½	2	3	3	2	2	2½	2	2½	2	2	26
1974 Traffic Model		1	-	1	2	1	2	1	2	2	1	2	2	17
Solar Physics														
SSPDA Model		1	1	1	1½	2	2	1½	2	1½	2	1½	2	19
1974 Traffic Model		-	1	1	2	2	2	2	2	2	2	2	2	20
Atmospheric & Space Physics														
SSPDA Model		-	1	1	1	3	3	4	3	4	3	4	3	30
1974 Traffic Model		-	1	1	2	2	2	2	2	2	2	2	2	20
Earth Observation & Earth & Ocean Physics														
SSPDA Model		3	3	3	3	2½	2½	3	3	3	3	3	3	35
1974 Traffic Model		-	-	2	2	3	2	3	3	4	4	5	6	34
Life Sciences														
SSPDA Model		-	1	1	1	1	1	1	1	1½	1½	1½	1½	13
1974 Traffic Model		-	1	1	2	2	2	2	2	2	2	2	2	20
Space Processing & Space Technology														
SSPDA Model		3	6½	7	6½	7	7	7	7	7	7	7	7	79
1974 Traffic Model		-	1	3	4	7	7	7	8	8	8	9	10	72
Multidiscipline														
1974 Traffic Model		1	1	1	-	-	1	-	1	-	1	1	1	8
Totals														
SSPDA Model		10	16	17	20	24	24	26	22	24	22	23	21	249
1974 Traffic Model		2	6	12	17	19	21	21	24	24	24	27	29	226

2.2.2.1 Spacelab Payload Development Sequence

Because the latter phases of the sequence of operations involved in a typical Spacelab flight are better defined, it is more convenient to discuss the sequence in reverse order. After the flight is over, the post-flight operations will include a post-flight calibration of the instruments followed by removal of the payload from the Spacelab. Following payload disassembly, the NIM/CAMAC units would be tested, maintenance or recalibration performed, as required, and they would be returned to a storage area, ready for their next assignment. We estimate that this sequence would require about three months at the most. It is probably desirable to delay disassembly of the instruments until at least some data analysis has been performed.

The flight itself lasts for seven to thirty days with the vast majority of flights being seven days.

The pre-flight integration period is essentially composed of four phases. In the first phase (Level IV) each experiment or instrument is integrated with the racks and pallet segments that it requires and qualification tests performed. This phase should require about five months. In the next phase (Level III) the combination of instruments composing the payload are integrated and checked out. In Level II the payload integration and checkout is extended to include the Spacelab flight subsystem support elements. Finally, Level I is composed of integration and checkout of the Spacelab with the Shuttle Orbiter. Level III through I integration is currently estimated to take one month. The total integration time for Levels IV through I is therefore about six months.

The payload equipment to be used in the flight, including the NIM/CAMAC equipment, will definitely have to be committed to the payload for the duration of the operations discussed up to this point. Thus the minimum period for which the flight NIM/CAMAC equipment must be committed is about nine months.

The payload development activities prior to Level IV integration will involve the design, development and testing of the individual instruments by the organizations responsible for the experiments or their contractors. We estimate that this activity will typically require at least nine months

and, frequently, several years may be needed. The existence of commercial units that are functionally identical to the NIM/CAMAC equipment to be flown, allows an approach to the phases of instrument development sequence prior to Level IV integration that may be very cost-effective. Normally the approach used during instrument development and testing prior to integration of the actual flight hardware involves a sequence of progressively more flight-like hardware such as breadboards, prototypes, etc. This development hardware is often nearly as expensive as the flight hardware. However, in the case of the NIM/CAMAC equipment, commercial units could be used if they are functional counterparts of the modified versions to be used in flight. We will consider the potential benefits of using commercial NIM/CAMAC equipment during instrument development in the next section.

2.2.2.2 Commercial NIM/CAMAC Equipment Use for Instrument Development

The use of commercial NIM/CAMAC units during instrument development and testing offers several advantages with respect to the use of equipment that has been modified and qualified for flight use.

Cost - Since the instrument development and testing prior to Level IV integration will require nine months or longer, the use of flight equipment during this phase of the instrument development would at least double the length of time that the flight units would have to be committed to a particular payload. Consequently, the amount of flight equipment needed in a pool to support the Spacelab payloads would also be about doubled. The cost benefit to be derived from using commercial counterparts during instrument development, and hence reducing the number of flight units needed by about one-half, can be assessed by recalling the modification costs discussed in Section 1.4.2. Considering recurring unit costs only, NIM/CAMAC modules suitably modified for flight are estimated to be two to seven times as expensive as commercial units, depending on whether the modifications are minimum or more extensive (see Tables 1-7 and 1-8). Therefore, the total NIM/CAMAC equipment costs will be reduced from 25 to 43 percent compared to the approach in which flight units are used throughout the instrument development sequence.

Maintainability - Since the instrument development activities prior to Level IV integration will be carried out by a variety of organizations at many different places, maintenance of the flight-qualified status of flight

equipment will pose a problem. If the flight modules were used for instrument development and testing, they would require a flight status certification after the return of the units from instrument testing, and prior to integration into the Spacelab. In contrast, if commercial units are used prior to Level IV integration, the flight modules would never be more widely dispersed than to the payload Level IV integration centers and certification of their flight qualification could be much more easily controlled and maintained. Any instrument qualification testing at the developer's facilities will now represent a qualification of the instrument sensor systems only. The full instrument qualification would occur during Level IV integration.

Flexibility - During the instrument development phase it is unlikely that the modules that were originally designated for the instrument will be a perfect selection. Changes in this complement of modules will be more difficult if flight modules are used. The paperwork, delay in receiving the new units and recertification of the old units will take time and cause inconvenience to the development program. If commercial modules were used for the instrument development, changes could be made quickly as the restrictions on their use should be minimal due to their low cost and non-flight status.

On the basis of these factors, the use of commercial units for instrument development and testing is clearly preferred. The availability of this option arises naturally in the case of NIM/CAMAC equipment due to the existing wide range of commercial units. Consequently, in our further discussion of NIM/CAMAC implementation for Spacelab payloads, we will assume that commercial units are used prior to Level IV integration and that the flight units will be committed to a given payload for a particular flight for a period of nine months.

2.2.3 NIM/CAMAC Equipment Usage Projections

Having established the baseline position that the flight equipment will need to be committed for about nine months for any flight, we can return to estimating the NIM/CAMAC equipment usage across the baseline Spacelab flight traffic model. The nine-month commitment period means that the equipment usage can be projected on an annual basis, i.e., on the average, equipment for payloads flying during any given year will have to

be committed to those payloads only during the flight year. On the other hand, the schedule margin is close enough to preclude the use of the same equipment for more than one flight in a year.

Hence, a reasonable estimate of the project annual NIM/CAMAC usage is given by simply multiplying the equipment requirements for the representative payloads by the number of flights per year given in Table 2-2 for the corresponding discipline. The NIM and CAMAC equipment requirements for the representative payloads were taken from Tables 3-53 and 3-54 in Volume II of this report. In the disciplines where results from more than one representative payload were available (astronomy, high-energy astrophysics and space physics) an average of the requirements for the available payloads was used. For multidiscipline payloads an overall average of the requirements from the numerous disciplines was used. The resulting overall NIM and CAMAC equipment usage per year, obtained by summing over the various types of modules and the different disciplines, is presented in Table 2-3.

Table 2-3. NIM/CAMAC Equipment Usage

<u>Year</u>	<u>Flights</u>	<u>Equipment Usage (Units/Year)</u>		
		<u>CAMAC</u>	<u>NIM</u>	<u>Total</u>
1980	2	156	210	366
1981	6	425	141	566
1982	12	645	324	969
1983	17	965	478	1443
1984	19	990	366	1356
1985	21	1066	507	1573
1986	21	1096	376	1472
1987	24	1156	523	1679
1988	24	1218	387	1605
1989	24	1071	523	1594
1990	27	1255	398	1653
1991	29	1292	548	1840

The annual usage rises up through 1983 as the number of flights per year increases to reach a level that remains fairly constant for the rest of the period covered by the baseline flight traffic model. The average annual usage from 1983 onward is about 1125 CAMAC units per year and 450 NIM units per year. It should be emphasized that these numbers are the projected usage in contrast to the annual number of units that must be procured. The procurement requirements could only equal the usage rate in the unlikely event that each unit was only used once. Before turning to the question of procurement requirements, we will investigate the characteristics of an equipment pool that could support the projected usage.

2.3 NIM/CAMAC EQUIPMENT POOL ANALYSIS

In any pooled equipment approach the following functions would be the responsibility of the pool organization:

- Procurement of equipment
- Distribution of equipment to users
- Maintenance and calibration of equipment
- Provision of technical information and support to users.

The main question that needs to be addressed is what type of equipment pool organization would perform these functions most efficiently and cost-effectively for the case of NIM/CAMAC equipment to be used in Spacelab payloads.

An obvious starting point is to draw upon the experience gained with NIM/CAMAC equipment pool operations at ground-based laboratories. However, there are significant differences between these examples and the situation that will apply for Spacelab payloads. In the typical case of the larger high-energy physics laboratories such as the National Accelerator Laboratory, Lawrence Berkeley Laboratory, Stanford Linear Accelerator Center, etc., the equipment users are located at the facility during the performance of their experiments. In this relatively simple situation where all of the demand for equipment comes from local sources, one central pool at each facility is clearly the answer. In the case of Spacelab payload operations, during experiment development the experimenters will be widely scattered throughout the U.S. and conceivably the world. As payload integration proceeds the activities will become progressively more centralized,

culminating in final integration at the Shuttle flight center (either Kennedy Space Center or Vandenberg Air Force Base). Therefore, a different pool organization may be more efficient in this case.

2.3.1 Alternative Pool Concepts

In principle, a wide spectrum of equipment pool concepts ranging from one centralized pool supporting all Spacelab users to a number of pools that each support a particular segment of the user community such as experimenters in one discipline or one geographical area could be considered. However, a centralized pool will always be more cost-effective in direct terms because of the following factors:

- Less duplication of effort and more efficient utilization of pool manpower
- More uniform demand for services due to averaging over a larger community of users
- Higher level of consolidation in equipment procurement.

Therefore, we have taken the approach of starting from the concept of one centralized pool and attempting to identify what requirements, if any, could justify the adoption of a more decentralized pool organization.

The disadvantages that normally arise from overcentralization mainly involve a loss of flexibility to deal with special user requests, inability to respond to demands for rapid service from a widely dispersed user community and intolerance to ill-defined or frequently changed user requirements. If any of these circumstances apply during the development and integration of Spacelab payloads, the resulting user inconvenience and delays could translate into cost increases that offset the cost advantages of a centralized equipment pool.

In considering the case of NIM/CAMAC equipment for Spacelab payloads, we could identify only a limited number of potentially significant problems that might occur with a centralized pool organization.

In terms of the functions of the pool organization, the procurement of flight NIM/CAMAC equipment and the provision of technical information and support would definitely be more efficiently handled by a centralized

organization. In addition, if the use of flight NIM/CAMAC equipment is limited to Level IV and higher integration phases as discussed in the preceding section, no problems were foreseen with maintenance and calibration of the flight equipment by a centralized organization.

Problems could arise with a centralized pool organization in the area of equipment distribution if a requirement for rapid service to widely dispersed users occurred. However, during the phases of the Spacelab payload operations sequence when rapid response is critical, i.e., during Level II, II and I integration, the users will all be located at one of the two flight centers. Thus, if the equipment pools are also located at the flight centers, delays would be held to a minimum.

During instrument development and testing prior to Level IV integration the users are widely dispersed and user requirements may also be ill-defined and frequently changed. However, in this case only commercial units would be involved. The distribution of commercial units could be handled by the central pool without restrictive controls because of their relatively low cost and the absence of any requirements to maintain a flight-qualified status. The instrument developers could also have the option of going directly to commercial suppliers if the central pool was unable to provide adequate service.

No meaningful comparison can be made at this time between the total inventory of equipment that must be maintained to support payload usage with alternative pool concepts. More data is needed than is available with the limited sample of representative payloads on the variation of equipment requirements from payload to payload in the segments of the user community served by individual pools in a multiple pool approach. In general, a centralized pool would be expected to require less equipment because of the averaging over a larger number of users. A rough indication of the differences that can be anticipated is provided by a comparison between the total equipment procurement requirements for a centralized pool to be calculated in Section 2.4 and for a dedicated equipment approach to be calculated in Section 2.5. As we will see, the dedicated approach, which to a certain extent represents a maximally decentralized approach, requires about 33 percent more equipment over the first six years of Spacelab operations.

In summary, we do not foresee any factors in the Spacelab user requirements for NIM/CAMAC equipment that could offset the overall cost advantages of a centralized pool approach if commercial equipment is used during the instrument development phase and provision is made for rapid response service at Vandenberg when Spacelab payloads start operating from there.

2.3.2 Recommended Pool Concept

The recommended pool concept is shown diagrammatically in Figure 2-1. The organization includes a control center for pool operations and equipment pool and distribution centers at the Shuttle flight centers (KSC and VAFB). The control center is a permanent element that performs the functions of overall pool management, technical information support for users and equipment procurement. The control center would most likely be located

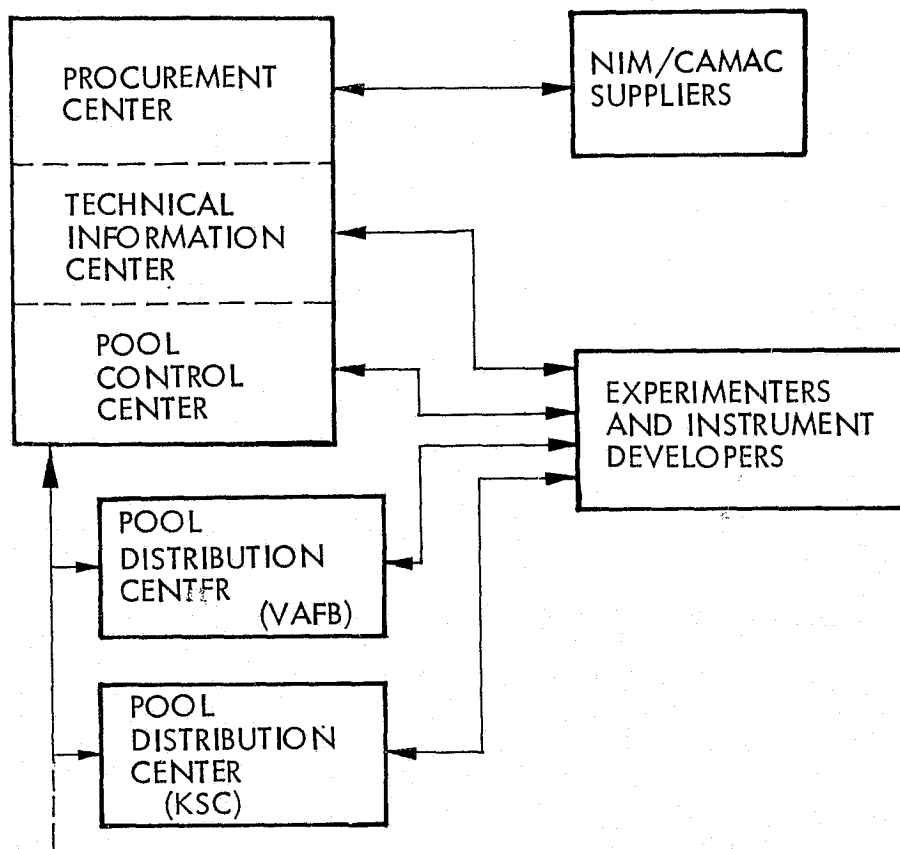


Figure 2-1. Recommended Organization of the NIM/CAMAC Equipment Pool

at KSC but this choice is not critical. The actual number of distribution centers in operation at any time will depend on the demand. For example, in the initial years of Spacelab operations only the distribution center at KSC will be needed since Shuttle operations will not start at Vandenberg until 1983. Additional distribution centers can be established if the sufficient demand exists.

The estimated manpower required to operate this pool and approximate annual labor costs at current rates are given in Table 2-4 for both the initial phase of operation in 1980 and for 1984 when equipment usage has reached a stable level and the pool is at its full size. The operational costs of the pool have to be considered when estimating the total cost of implementing a shared equipment approach for NIM/CAMAC equipment. However, most of the functions provided by the pool will have to be provided by somebody in any approach, so only a small fraction of the pool operational costs can be uniquely associated with the use of an equipment pool.

Table 2-4. Manpower Requirements and Costs for
Operation of the Recommended Pool Concept

<u>Location</u>	<u>Labor Category</u>	<u>Number of Persons and Cost/Year (\$000)</u>	
		<u>1980</u>	<u>1984</u>
Control Center	Manager	1 @ 65 = 65	1 @ 65 = 65
	Procurement Officer	1 @ 50 = 50	1 @ 50 = 50
	Information Officer	2 @ 50 = 100	2 @ 50 = 100
	Clerical & Support	2 @ 25 = 50	3 @ 25 = 75
KSC Pool	Supervisor	1 @ 50 = 50	1 @ 50 = 50
	Technician	1 @ 40 = 40	3 @ 40 = 120
	Clerical & Support	1 @ 30 = 30	3 @ 30 = 90
VAFB Pool	Supervisor	-	1 @ 50 = 50
	Technician		1 @ 40 = 40
	Clerical & Support		1 @ 30 = 30
Total Cost		\$385K	\$670K

2.4 EQUIPMENT COSTS AND POOL MANAGEMENT PLAN

Before NIM/CAMAC equipment costs can be estimated for the pooled-equipment approach, the time-phased procurement requirements to support the estimated yearly usage shown in Table 2-3 must be determined.

2.4.1 Pool Equipment Procurement Requirements

The minimum pool equipment procurement requirements were calculated directly from the detailed equipment usage requirements as follows: starting with the tabulation of the numbers of each type of NIM and CAMAC equipment item used in each year by the payloads in each discipline (i.e., the number of flights in each discipline for each year times the appropriate representative payload equipment requirements), the annual usage requirements for each NIM and CAMAC equipment item were determined by summing over all of the disciplines. The annual procurement requirement for each equipment item is equal to the number of units that must be added to the pool each year to maintain an inventory that is at least equal to the number of units to be used in the year. The results of this calculation for the first four years of Spacelab operations (the period of primary pool buildup) are tabulated in Table 2-5. These minimum procurement requirements were next adjusted upwards to take into account the needs for spare units and replacement units.

2.4.1.1 Space Unit Requirements

We assumed that a number of spare units approximately equal to twenty percent of the number of units in the pool should be available to cover contingencies and variations in user requirements. However, because of the small size of the pool in the early years, the procurement profile was adjusted to provide close to forty percent spares in the first year and a gradual decline in the number of spare units to an average of about fifteen percent when the pool has reached full size. This approach has the added advantage of smoothing out the fluctuations in the yearly procurement profile.

2.4.1.2 Replacement Unit Requirements

Since the equipment has a finite life expectancy, new units will have to be procured as old units are removed from inventory. The effective life

Table 2-5. Equipment Pool Procurement Requirements for 1980-1983
(no spares or replacements included)

CAMAC Equipment Item	CAMAC Code	Year				NIM Equipment Item	Year			
		80	81	82	83		80	81	82	83
Scalars	111	23	0	13	23	Shaping Amplifiers	37	7	25	35
Preset Scalars	113	1	4	3	3	Fast Amplifiers	2	0	0	0
Position Encoders	117	6	8	8	12	Delay Amplifiers	3	0	0	1
Input Gates	121	6	15	15	11	Sum/Invert Amplifiers	32	0	1	19
Input Registers	122	1	6	8	0	Linear Gates	1	0	16	0
Logic Units	123	4	0	2	2	Fast Linear Fan-Ins	9	0	0	3
Interrupt Registers	127	2	4	3	5	Fast Linear Fan-Outs	3	0	0	1
Clocks & Pulse Generators	131	4	6	2	12	Fast Integral Discriminators	21	0	0	11
Output Registers	132	5	12	14	16	Slow Integral Discriminators	6	9	3	12
Output Drivers	133	11	23	30	20	Single Channel Analyzers	1	3	7	6
Stepping Motor Controllers	145	16	42	26	44	Zero Crossing Discriminators	5	0	0	3
Analog to Digital Converters						Constant Fraction Discriminators	8	0	0	4
High Resolution - Fast	161	18	56	12	48	Coincidence Units	15	0	0	7
Multichannel - Slow	161	15	40	25	37	Pulse Height Analyzer	1	1	0	0
Time Digitizers	161	16	0	0	12	High Voltage Power Supplies	38	0	34	38
Digital to Analog Converters	162	4	15	6	14	NIM Bins W/Power Supply	21	0	11	6
Multiplexers	164	4	23	0	9	Special Modules				
Branch Drivers	211	2	5	4	5	Sequence Discriminator	5	0	4	7
Crate Controllers	231	9	19	14	21	Wave Analyzer	1	3	0	2
Crates W/Power Supply	411	9	19	14	21	Differential Amplifier	1	0	0	0
CAMAC Totals		156	297	199	315	NIM Totals	210	23	101	155

expectancy of NIM/CAMAC equipment being used in Spacelab payloads is a difficult quantity to estimate. The possible factors to be considered in arriving at an estimate are:

- Failure rates
- Maintenance costs vs. replacement costs
- Obsolescence

Conventional failure rates for NIM/CAMAC equipment can be estimated from failure rate data on the types of electronic components used in the circuits. Even for industrial-grade parts, the corresponding life expectancies are greater than ten years. In addition, the units can be repaired so the calculated failure rates don't represent the actual situation. The real failures will probably occur because of overstressed components, marginal design, or misuse. Recalling the discussion of the modifications for Spacelab use in Section 1.4, the first two causes are much more likely to occur in equipment with minimum modifications since the commercial circuit design and part selection are used without modification. For this type of failure, repair by simply replacing the failed part does not cure the problem. In any case, the life expectancy due to actual failures will probably be on the order of five to ten years for minimum-modified equipment and greater than ten years for more extensively modified equipment.

Another limit on the effective life of the equipment is the point at which it becomes more expensive to continue maintaining a unit than to replace it. We estimate the typical maintenance cycle for a unit will cost about \$200. Hence, at a rate of one cycle per year, even the minimally modified units could be maintained for six years before the maintenance costs equal the original cost.

The factor that we believe will really control the effective useful life of the equipment is obsolescence. If new units with improved performance are available, users will tend to quit using the older models. The situation is actually a tradeoff between the increased costs of a higher replacement rate and the users' preference for the latest model. The most cost-effective approach will be for the pool to resist the users' tendency to switch to a new model unless it is truly required. Experience with NIM/CAMAC equipment pools indicates that the useful life of a unit in these circumstances is about seven years, on the average. We, therefore, have

used this number for the life expectancy of a more extensively modified unit. In view of the lower unit cost and higher expected failure rate of minimally modified equipment, we have assumed four years to be its average life expectancy.

The average life expectancies were factored into the equipment procurement requirements by using an annual replacement rate for pool equipment that results in total replacement of the inventory in a period equal to the average life expectancy.

2.4.1.3 Refined Pool Equipment Procurement Requirements

The annual NIM/CAMAC equipment procurement requirements for the recommended pool concept, which were calculated as described in the preceding sections, are given in Table 2-6.

Table 2-6. NIM/CAMAC Equipment Pool Procurement Requirements

<u>Year</u>	<u>Pool Size</u>	<u>NIM/CAMAC Units Procured per Year</u>	
		<u>Minimum Modification</u>	<u>More Extensive Modification</u>
1980	500	500	500
1981	785	410	356
1982	1200	611	527
1983	1720	820	691
1984	1720	430	246
1985	1800	510	326
1986	1800	450	257
1987	1850	500	307
1988	1850	463	264
1989	1850	463	264
1990	1850	463	264
1991	1850	463	264

2.4.2 Cost Estimate for Pool Equipment

NIM/CAMAC pool equipment costs were estimated on the basis of both minimally modified and more extensively modified flight units. The non-recurring design, development and test costs as well as the recurring unit

costs were given in Tables 1-7 and 1-8. For our purposes here, these estimates were used to generate the average module cost including non-recurring development as a function of the number of units procured. This relationship is plotted in Figure 2-2 for both modification cases. The cost estimates for the NIM/CAMAC pool equipment were then generated using annual procurement requirements for each type of NIM and CAMAC equipment item (the equivalent of Table 2-5 after spare and replacement units were added) and the curves in Figure 2-2. The results are given in Table 2-7. Two significant points should be noted. First, the relatively low cost of the pool equipment in general, especially after the initial buildup period. Second, although the more extensively modified units are initially four to five times as expensive as the minimally modified units, the difference over the entire 1980-1991 period is only a factor of two due to the large quantity of units procured.

These cost estimates cover only the NIM/CAMAC flight equipment. An equal quantity of commercial units will need to be procured for instrument development and testing. The total cost of commercial units over the twelve-year period is about \$3.6 million or an average of \$0.3 million/year. Although significant, this cost is well below the flight unit costs. Also the pool operational costs given in Table 2-4 should be included for a more complete NIM/CAMAC pool cost estimate. The cumulative pool operational cost over the twelve-year period is \$8.0 million. Including the commercial units and pool operational costs, the total pool cumulative costs for 1980-1991 are \$20.8 million and \$30.6 million, respectively, for the minimum and more extensive modification cases. Hence, the cost difference between the two levels of modification is even less significant when the fixed overhead of the equipment pool is taken into account.

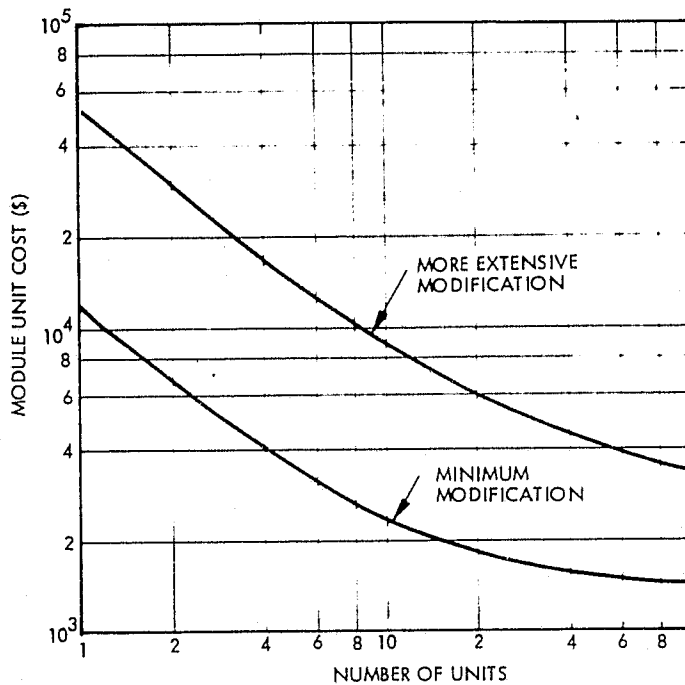


Figure 2-2. Average Module Costs Including Nonrecurring Development Versus Numbers of Units Produced

Table 2-7. Estimated Costs for NIM/CAMAC Pool Equipment

Year	Equipment Costs (M\$)			
	Minimum Modification	More Extensive Modification	Cost/Year	Cumulative
	Cost/Year	Cumulative	Cost/Year	Cumulative
1980	1.15	1.15	4.25	4.25
1981	0.66	1.81	1.71	5.96
1982	0.92	2.73	2.30	8.26
1983	1.23	3.96	2.59	10.85
1984	0.60	4.56	0.92	11.77
1985	0.71	5.27	1.22	12.99
1986	0.63	5.90	0.96	13.95
1987	0.70	6.60	1.15	15.10
1988	0.65	7.25	0.91	16.01
1989	0.65	7.90	0.99	17.00
1990	0.65	8.55	0.99	17.99
1991	0.65	9.20	0.99	18.98

2.5 DEDICATED EQUIPMENT APPROACHES

Our primary objective in considering implementation approaches in which the NIM/CAMAC equipment used by each payload was assumed to be dedicated to that payload and not available to other users was to provide a cost comparison with the pool approach. In addition to providing the basis for the determination of the cost impact of equipment sharing by users, consideration of a dedicated equipment approach also allowed us to make a cost comparison between the use of standard NIM/CAMAC equipment and the use of functionally equivalent, custom-built equipment. This type of equipment would by definition be dedicated to the payload for which it was developed.

2.5.1 Equipment Requirements for Dedicated Approaches

It is important to realize that even in a dedicated equipment approach, the equipment will in general be used a number of times if the payload to which it is dedicated is reflown. The relative cost of dedicated equipment usage compared to the adoption of an equipment pool thus depends critically on the number of payload reflights. This can be seen by considering the situation in which a payload using dedicated equipment is flown every year. In this case, it makes no difference if the equipment is dedicated to the payload or is part of an equipment pool. Therefore, it is necessary to establish the actual new payloads in our baseline Spacelab flight traffic model as opposed to reflights of existing payloads.

2.5.1.1 Baseline Model for New Payloads

The number of new payloads in the baseline Spacelab traffic model was estimated on the basis of the mission frequency information contained in the 1974 version of the SSPDA documents. The method used was similar to that previously described in Section 2.2.1. In this case, however, an instrument or payload listed in the SSPDA tabulation was only included in the calculation of the number of full payloads in each discipline for the first year in which it appeared. In other words, reflights were not counted. This information was converted to the number of new payloads in the baseline model by assuming that the fraction of the total number of flights per year in each discipline which were new payloads was the same as that found in the SSPDA tabulations.

The result of this process was that only 23 out of a total of 77 flights in the first six years of the baseline model (1980-1985) involved new payloads and essentially all flights after 1985 were reflights. Naturally, this simply reflects the understandable fact that there is a very large number of reflights projected in the SSPDA tabulation and all of the payloads identified are assumed to fly for the first time before 1985. This is undoubtedly not a realistic representation of Spacelab payload operations, but no better information is available on which to base the estimate. Since the number of new payloads estimated in this way becomes increasingly unrealistic in the later years of the payload model, we only carried the exercise on dedicated equipment through 1985.

2.5.1.2 NIM/CAMAC Equipment Procurement Requirements for a Dedicated Approach

Given the number of new payloads in the baseline model, it is a straightforward process to estimate the amount of equipment that must be procured each year in a manner that is analogous to that used for the pooled equipment case. The same assumptions were used to adjust the initial estimated requirements to take into account the finite life expectancy of the equipment and the need for spare units to cover contingencies (i.e., replacement cycles of four and seven years, respectively, for minimum and more extensive modifications; and 20 percent spare units). The resulting total numbers of units that must be procured each year are given in Table 2-8.

Table 2-8. Dedicated NIM/CAMAC Equipment Procurement Requirements

<u>Year</u>	NIM/CAMAC Units Procured per Year			
	<u>Baseline Traffic Model Flights</u>	<u>New Payloads</u>	<u>Minimum Modification</u>	<u>More Extensive Modification</u>
1980	2	2	383	383
1981	6	6	656	615
1982	12	8	1064	963
1983	17	4	888	698
1984	19	1 1/2	645	408
1985	21	1 1/2	705	458

The total procurement requirement over the 1980-1985 time period is only about 33 percent greater than in the pool approach for both minimum and more extensive modifications. The advantages of the pool approach would probably be more significant in the following years. Whereas the pool has attained full size by 1985 and only requires a reasonable amount of replacement to be maintained, new requirements will continue to arise in the dedicated equipment case. A reasonable approximation of the 1986-1991 time period for the dedicated case would probably be given by assuming a repeat of the 1980-1985 requirements. If so, the dedicated equipment approach would require procurement of about twice as many modules as the pool approach.

2.5.2 Estimated Costs for Dedicated Approaches

2.5.2.1 Dedicated NIM/CAMAC Equipment Costs

The cost estimates for the dedicated NIM/CAMAC equipment were generated in the same way as for the pool equipment. In particular, it was assumed that although the equipment would be dedicated to individual payloads, the procurement requirements for the various payloads would be consolidated. Even if this assumption was not literally valid, the unit prices of the equipment would certainly reflect the overall level of procurement. The results are given in Table 2-9.

2.5.2.2 Custom-Built Equivalent Equipment Costs

It is also a straightforward process, given the number of new payloads in each discipline, to estimate the comparable cost of implementing the payloads in the way it is conventionally done at present. For this case, equipment that is functionally equivalent to the NIM/CAMAC equipment required by the payload would be developed and manufactured specially for each payload. It was assumed that advantage would be taken of the commonality of requirements that existed within each payload. Thus, a unit cost versus number of units curve, analogous to those in Figure 2-2, but based on Table 1-9, was used to estimate the cost of the equipment required for each representative payload plus 20 percent spare units. No equipment replacement was included in the estimated cost. The costs of the representative payload in each discipline were simply multiplied by the numbers of new payloads in the disciplines to arrive at the programmatic

Table 2-9. Equipment Costs for Dedicated Approaches

<u>Year</u>	<u>Equipment Costs (M\$)</u>					
	<u>Minimum Modification Cost/Year</u>	<u>Cumulative</u>	<u>More Extensive Modification Cost/Year</u>	<u>Cumulative</u>	<u>Custom-Built Cost/Year</u>	<u>Cumulative</u>
1980	0.88	0.88	3.26	3.26	8.66	8.66
1981	0.98	1.86	2.74	6.00	16.60	25.26
1982	1.49	3.35	3.35	9.35	24.40	49.66
1983	1.24	4.59	2.15	11.50	12.65	62.31
1984	0.90	5.49	1.26	12.76	3.80	66.11
1985	0.99	6.48	1.41	14.17	4.56	70.67

cost estimates. The results are given in Table 2-9 along with those for modified NIM/CAMAC equipment.

2.5.3 Comparison of Costs for the Alternative Approaches

Because of the low number of new payloads in the baseline payload model used and the assumption of consolidated procurement, the estimated cumulative equipment costs for a dedicated implementation approach are only slightly greater than the comparable pool equipment costs (\$6.5 million versus \$5.3 million and \$14.2 million versus \$13.0 million, respectively, for the minimum and more extensive modification cases).

Although the frequency of reflights has probably been overestimated, these results do indicate that the cost saving to be realized by the establishment of an equipment pool will probably not be great in the early years of Spacelab payload operations. As already discussed, the cost benefits of a pooled-equipment approach will probably be more significant in the later years of payload operations. This suggests that the most reasonable approach would be to start out by setting up the amount of central control needed to establish the standards to which equipment is to be built and to coordinate the equipment requirements and procurements of the various payloads, but to not set up an actual equipment pool. As the situation evolves, actual equipment pool operations can be initiated when warranted by the level of equipment usage and degree of user acceptance.

In contrast to the relative costs of pooled and dedicated approaches, the cost of comparable custom-built equipment is seen to be five to ten times greater than the implementations using standard NIM/CAMAC equipment. While this is due in part to the higher nonrecurring development and recurring unit costs used for this equipment, the largest portion of this cost increase is due to the assumed absence of standardization beyond the payload level. In other words, the amortization of nonrecurring development costs is greatly reduced in this case.

In order to illustrate this point, our analysis of alternative implementation approaches can be generalized to consider all of the available options. The generalization can be viewed as an investigation of the equipment costs as a function of two independent parameters: the level of equipment modification and the degree of standardization in the implementation approach.

So far as the level of modification is concerned, we already have the necessary information for three levels, if what we have been using to this point as the approach for custom-built equipment (see Section 1.4.2.3) is interpreted as the maximum level of modification. The estimated costs (Table 1-9) do not depend on whether the equipment is standard or custom-built.

Two key questions are involved in the alternative implementation approaches: is the equipment shared by users or dedicated to individual payloads and is the equipment procured in common for all payloads or procured separately for each payload? Out of the four possible combinations, the case of separate procurement but shared usage is not germane. Thus three alternatives are available.

The comparable equipment costs for all nine possible options can easily be estimated using the methods and assumptions already described. In fact, five options have already been estimated. The estimated 1980-85 cumulative costs for the complete matrix of possible options are given in Table 2-10. The results illustrate the fact that the sharing of nonrecurring development costs made possible by standardization is more important than shared usage of the equipment. As would be expected, this conclusion becomes stronger as the level of modification, and hence the nonrecurring development cost, increases.

Table 2-10. Comparative Costs of Alternative Equipment Implementations

Equipment Implementation Approach	<u>Cumulative Equipment Costs for 1980-1985 (M\$)</u>		
	<u>Minimum</u>	<u>More Extensive</u>	<u>Maximum</u>
Pooled Standard Equipment	5.3	13.0	22.3
<ul style="list-style-type: none"> ● shared usage ● common procurement for all payloads 			
Dedicated Standard Equipment	6.5	14.2	26.1
<ul style="list-style-type: none"> ● dedicated usage ● common procurement for all payloads 			
Dedicated Custom Equipment	8.9	35.6	70.7
<ul style="list-style-type: none"> ● dedicated usage ● separate procurement for each payload 	79		

3. IMPLEMENTATION AND IMPACT OF CAMAC ON SPACELAB EXPERIMENT SOFTWARE (TASK 4B)

3.1 INTRODUCTION

In order to properly assess the cost effectiveness of adopting the NIM/CAMAC standards for Shuttle experiment data acquisition and control systems, one cannot ignore the associated software development and implementation costs. No cost savings can accrue to NASA if, in the effort to minimize expenditures, an economical hardware standard is adopted that requires extensive additional software expenditures cancelling or exceeding the savings originally derived from the hardware standards. If a CAMAC hardware system is implemented for experiment control and data management, significant portions of the software will directly support the hardware functions and since these hardware functions will be common to many experiments there will also be considerable commonality in the supporting software. Therefore, those portions of the experiment software systems that directly support the CAMAC hardware functions need be written only once and can be supplied to the individual experiments in parallel with the pooled hardware.

3.1.1 Scope of the Software Task

The objective of this task is to investigate the impact and implementation of such a software system. The effort was divided into four subtasks.

- Task 4B.1 Survey and summarize representative existing CAMAC software systems.
- Task 4B.2 Survey and summarize current information on the Spacelab software system.
- Task 4B.3 Investigate a system of pooled CAMAC support software.
- Task 4B.4 Analyze the major software requirements for two representative payloads.

Four existing CAMAC software systems were selected from the available examples. These four systems provide a reasonable sample of the range of CAMAC software system concepts used in different applications that each have at least some key requirements that will be encountered in implementing Spacelab payload software. All of the available documentation on these software systems was obtained and a summary of the relevant features of each was prepared with an emphasis on their approach to user application program

implementation.

Next, the available documentation describing the Spacelab software environment for payloads was reviewed and summarized with emphasis on those features most relevant to payloads using CAMAC hardware.

The results obtained in the surveys of existing CAMAC software systems and the Spacelab software system were applied to investigate software implementation for CAMAC systems used in Spacelab. Functional criteria were identified to distinguish two general categories of CAMAC usage in Spacelab payloads and recommended approaches to handle each were formulated. The types of standard CAMAC software to be provided for users were defined and the impact of the use of CAMAC hardware on experiment software development costs was assessed.

Finally, the major software requirements were analyzed for two of the representative payloads selected and analyzed in Tasks 1 and 2. Top level software system diagrams were developed to provide specific examples of the recommended approaches to CAMAC software implementation and the standard CAMAC interface subroutines required by each payload were identified.

3.1.2 General Experiment Software System Requirements

The following major elements are required in a software system to be used for experiment control and data acquisition:

- The operating system for the processor which handles executive services such as task scheduling, system resource allocation and system initialization and loading.
- Input/output drivers which handle data transfers to and from peripheral hardware.
- A utility library which provides commonly-used computation and analysis routines, display control routines etc.
- The application program which defines the sequence of operations required by the experiment.
- Software development aids such as high-order language compilers, assemblers, editors and simulators.

The operating system of the software is unique to the computer central processing unit and is usually supplied by the computer manufacturer. The

operating system is a collection of programs such as the monitor program, executive program, system loader, system preparation routine; i.e., those programs required to allow the hardware to perform the desired functions of a computer. The operating system is written in efficient machine language.

The input/output drivers handle communications between the central processing unit and peripheral hardware such as tape recorders, printers, disc memories, keyboard units, display units, and of particular interest here, experiment data acquisition and control hardware such as CAMAC.

Most software systems include a library of utility routines that perform commonly used functions such as special mathematical functions, statistical analyses, matrix manipulation, display control, etc. These utility routines are usually designed to be called from a high-order language program and facilitate the user's development of his application program. The utility routines are frequently written in assembly language to maximize operating efficiency.

The application program is the software which has been created to provide the events and data acquisition desired by the experimenter. This must be accomplished within the constraints of the operating system and the hardware system. This software can be in a high-level language to minimize programmer time or in assembly language to minimize core requirement and/or minimize machine time.

The software development aids are all intended to minimize the user's effort required to generate, integrate, and check out his software.

All of these elements except the applications program, which must be developed for each specific experiment, are usually provided to the user by the host software system and certainly should be provided for Spacelab users. The magnitude of the experiment software effort depends critically on the availability and convenience of use of these software system elements. Ideally, the experiment software development should only involve developing the applications program.

3.1.3 Impact of CAMAC on Software Requirements

The use of CAMAC hardware really only directly impacts the software system by allowing the use of standard input/output drivers for the CAMAC

hardware. The drivers should ideally make the details of the host software system as transparent as possible to the user.

The CAMAC driver routine loads all the CAMAC commands, which are generally provided by the application program, into the registers of the CAMAC branch driver (serial or parallel) or of the crate controller if a stand-alone type U crate controller. Since bit manipulation and transfers to specific core addresses are often required, this must be accomplished by a low-level language. In addition, as the low-level language depends on the computer and the registers can differ among branch drivers, this portion of the software is usually unique for each computer and each manufacturer's branch driver. While used often, this routine really only transfers four standard CAMAC pieces of information; i.e., 1) CAMAC function, 2) CAMAC address, 3) CAMAC data (when required), and 4) CAMAC status.

The actual manipulation of CAMAC commands and data is accomplished differently by the various users as will be pointed out in the discussions of the four CAMAC applications which are reviewed in the following sections. Most CAMAC software implementations aimed at simplifying the creation of the application program are based upon a subroutine for each CAMAC module to construct the desired call to the CAMAC driver.

In considering the implementation of the Spacelab software system, it is evident that compared to completely unique systems for each experiment there are many potential advantages to working with a subsystem of CAMAC software interfacing with CAMAC hardware in the experiment control and data management system.

A major advantage of a CAMAC system is that the hardware/software interface is firmly established. This means that the details of the interface do not have to be readdressed each time the software for a new experiment is being generated. The experiment unique software has only to intelligently call the module level subroutines in order to communicate with the hardware. This results in a significant reduction in the amount of software that has to be written for each individual experiment.

Another advantage is that since the CAMAC system software need be written only once to handle all experiments, it can be written in the assembler language of the host system. This minimizes the core space

required by the subroutines and maximizes their efficiency. It also means that the CAMAC software can be documented more thoroughly as it is generated making it more accessible to new users and review personnel.

A final advantage of the CAMAC software system is the ease with which it is understood. Since it is structured to directly support the pool of hardware, there is never any indecision about the software requirements of a given hardware system. As soon as the hardware modules for an experiment are chosen, the software support functions required by these modules are known. The process of reviewing and approving an experiment software system for flight is greatly simplified since large sections of that software will be from the CAMAC software pool.

3.2 EXISTING SOFTWARE SYSTEMS

Four examples of software implementation for CAMAC systems were selected from the large number of available examples. As in the case of CAMAC hardware (see Section 3.1.2 and Appendix I, Volume II), a CAMAC Product Guide for software is published in each issue of the CAMAC Bulletin. The edition from Issue No. 14 (December 1975) is reproduced in Appendix I of this volume to illustrate the amount and type of CAMAC software currently available.

3.2.1 Hot Fuel Examination Facility

3.2.1.1 General Description

The Data Acquisition and Process Control System at the Hot Fuel Examination Facility is primarily a dedicated system for computerized automatic control and data acquisition of fuel element examination via two CAMAC parallel highways. The software system is relatively static because of its dedicated function. The software system, operating on a Datacraft 6024/3 central processor, makes extensive use of assembly language to achieve high operating efficiency.

Hot-cell examinations such as gamma scanning produce vast amounts of data, and only so much data can be taken during an eight-hour shift. Automation of the examination device has improved the situation by making it possible to operate the device twenty-four hours a day without operator attention and by improving the efficiency of the machine through automation.

The Data Acquisition and Process Control System uses a Datacraft 6024/3 central processing unit. This is a medium-sized computer with a word length of 24 bits, a cycle time of 1 microsecond, and 32-k-word magnetic core memory. Peripheral equipment consists of a 28-megabyte moving head disc, two 9-track magnetic tape units, four 7-track magnetic tape units, card reader, line printer, teletype, engineering display terminal, three remote terminals, and the two CAMAC parallel highway systems. Figure 3-1 is a schematic of the system.

3.2.1.2 Software System Description

Operation System - The Disc Monitoring System (DMS-III) operating system provides foreground multiprogramming concurrently with background batch processing. The real-time, application programs are run in foreground and

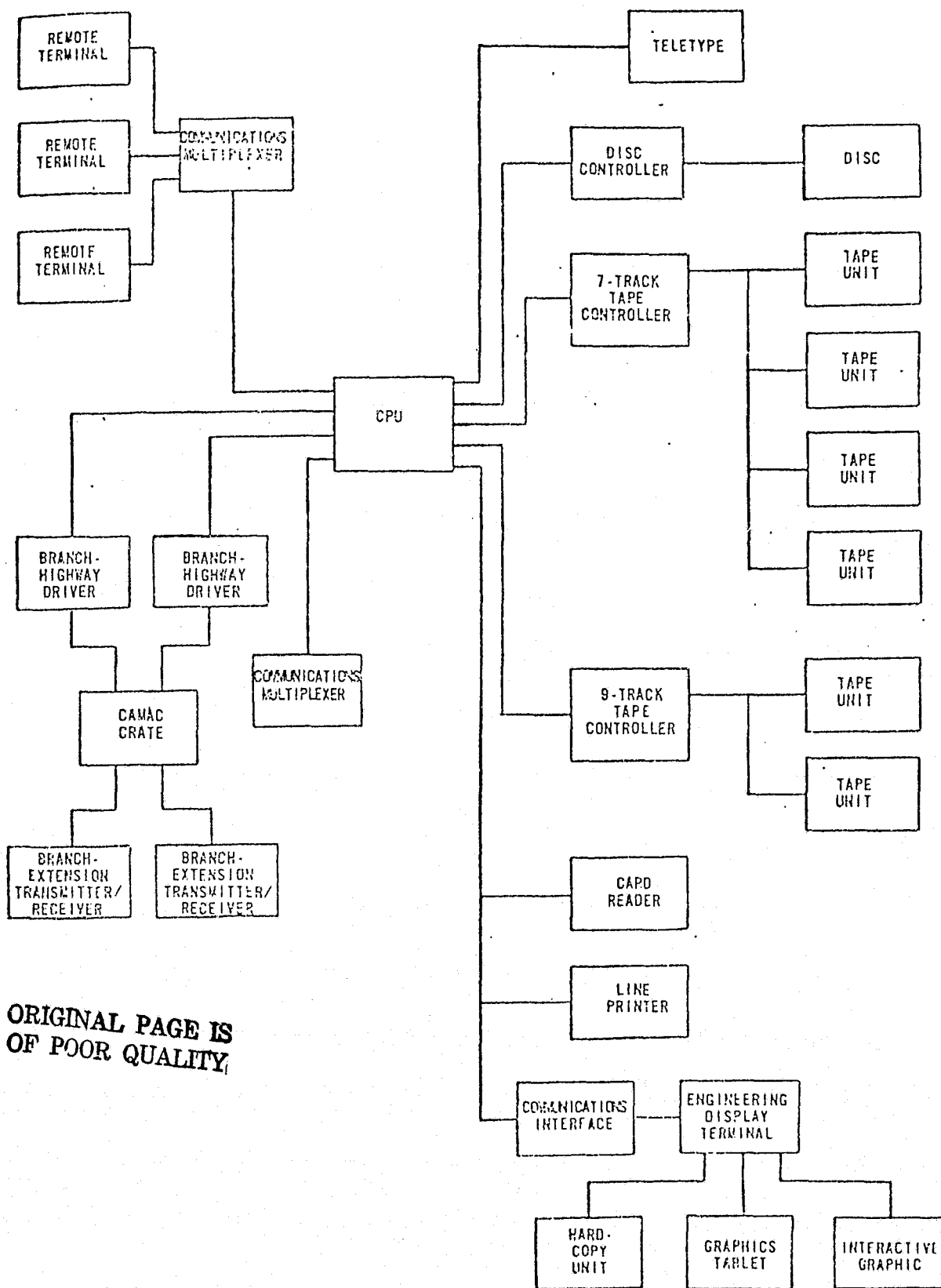


Figure 3-1. Hot Fuel Examination Facility Data Acquisition and Process Control System

receive highest priority. Background programs, which carry out the data reduction, are serviced as time permits. Each time a significant event occurs; e.g., an operation that causes an active program to become idle, the program list is examined from top to bottom to find a program to run, the dispatcher enters an idle loop to await another significant event; i.e., an operation that causes a suspended program to become active.

For high-speed devices such as magnetic tape or disc units, the transfer of data one word at a time under program control is too slow. The Datacraft operating system provides for Automatic Block Transfer Channel (ABC). After the ABC is initialized with information on size of block and storage locations, data is transferred twenty-four bits at a time without CPU action. An individual channel is capable of using one out of every three memory cycles for a rate of 333-k words per second. By means of special instructions, multiple channels can be overlapped to achieve an aggregate rate of one million words per second.

Application Programs - Programs for DAPCS have been designed in modular fashion; i.e., usually each CAMAC module has a corresponding subroutine. Where several CAMAC modules are similar, more generalized subroutines have been designed. These subroutines reside as members of the disc library files and can be accessed by any program.

The data word output from the software system to the CAMAC system is twenty-four bits in length. This is possible due to the compatibility of the computers 24-bit word and the standard CAMAC 24-bit word.

The six parts of the CAMAC command word are standardized as crate address (CR), station or module number (N), subaddress (A), function code (F), initialize (Z), and graded-L request (BG). These components of the CAMAC command word are arranged from the most significant bits to least significant bits (left to right) as follows: (BG), (Z), (F), (A), (N), and (CR).

The CAMAC command words are given mnemonics that correspond to the CAMAC task to be performed. For example, the mnemonics for the command word to read the ADC Multiplexer for the fuel element-clamping-guide force ($CR\ N\ A_0\ F_0$) is RADCCG. Broken down, this is Read (F_0) the ADC (N) for Clamping Guide force (A_0). Similarly, RADCSR refers to Read (F_0) the ADC

(N) for Sense Rod position (A_2). The command words and their mnemonics are stored in the subroutine or program where they are used.

The components (F A N CR) of the command word could be passed to the CAMAC driver with the actual command word put together in the CAMAC driver. This would eliminate the storage of command words; however, the same command words would have to be assembled many times and the CAMAC driver would be lengthy. As indicated in the previous paragraph, the command words are defined and stored in the program where they are used.

CAMAC Driver - The CAMAC driver for the DAPCS is a subroutine OUTWD. It is necessarily written in Datacraft assembly language. The efficiency afforded by the assembly language is also important because the CAMAC driver is called over 2-1/2 million times per Gamma Scan. In an average precision-gamma-scanning program, around 300 spectra are taken and OUTWD is called roughly 8300 times per spectrum.

The main features of the routine OUTWD as the means of implementing CAMAC commands are as follows:

- OUTWD provides a standard method of addressing all CAMAC systems. All real-time programs are written in a similar way. Training required for new programmers is reduced and programming time required for experienced programmers is decreased.
- The subroutine can be called from a main program written in assembly language or a higher-level language such as FORTRAN.
- Each time a command word is output to a remote crate, the on-line status of that crate is automatically checked.
- Handshake must be returned from the addressed crate before the program will continue.
- A second try to output the command word is attempted if the handshake is lost because of electrical noise.
- Transmission errors are checked for in either direction.
- A second try to output the command word is attempted if a transmission error occurs because of electrical noise. This includes parity, framing, and stop/start bit errors.
- The advantages of a subroutine such as OUTWD outweigh the overhead time in its execution. A normal execution of OUTWD takes approximately 100 machine cycles or 100 microseconds.

3.2.2 Fermi National Accelerator Laboratory

3.2.2.1 General Description

The Basic Instrument for the Support of On-Line Needs (BISON) system at Fermilab provides high-speed communication links using CAMAC equipment to interconnect two central CDC 6600's with a variety of user minicomputers (mostly DEC PDP-11's) in a multiplexed star network. The software system provides for efficient, transparent data transmissions between users and the central computers. For immediate data analysis, each experiment is allowed to transfer one 1024-word buffer from its minicomputer to the CDC-6600's, per accelerator cycle. The majority of the data are reduced and analyzed either by the controlling minicomputer with visual display or batch processed by the CDC-6600's.

The intercomputer communication is by CAMAC modules which provide hardware independence. Each station consists of a transmit/receive module and two 1024-word, 24-bit memories. One memory is the transmit memory and the other the receive memory. At the other end of two coaxial cables is a similar set of modules, and each set of modules is connected by a proper interface to a computer. This provides CAMAC-controlled computer-to-computer communication. This communication link is shown schematically in Figure 3-2. Figure 3-3 illustrates a typical PDP-11 BISON configuration.

3.2.2.2 Software System Description

The Fermilab has developed a library of software to support the CAMAC instrumented experiments. A number of PDP-11 operating systems are used. In addition to operating systems provided by Digital Equipment Corporation (DEC), Fermilab has developed their own PDP-11 operating systems called SPEX and BSX. Of course, each experimenter must develop his specific application program, but Fermilab has many subroutines to aid in debugging programs, handling data, displaying data, formatting, etc. FORTRAN callable CAMAC handlers or drivers in assembly language are provided for several branch driver/PDP-11 interfaces.

Fermilab has developed an interpreter to format desired CAMAC commands into Task tables that are then used by the CAMAC driver. This aids in the development of application programs by the experimenters.

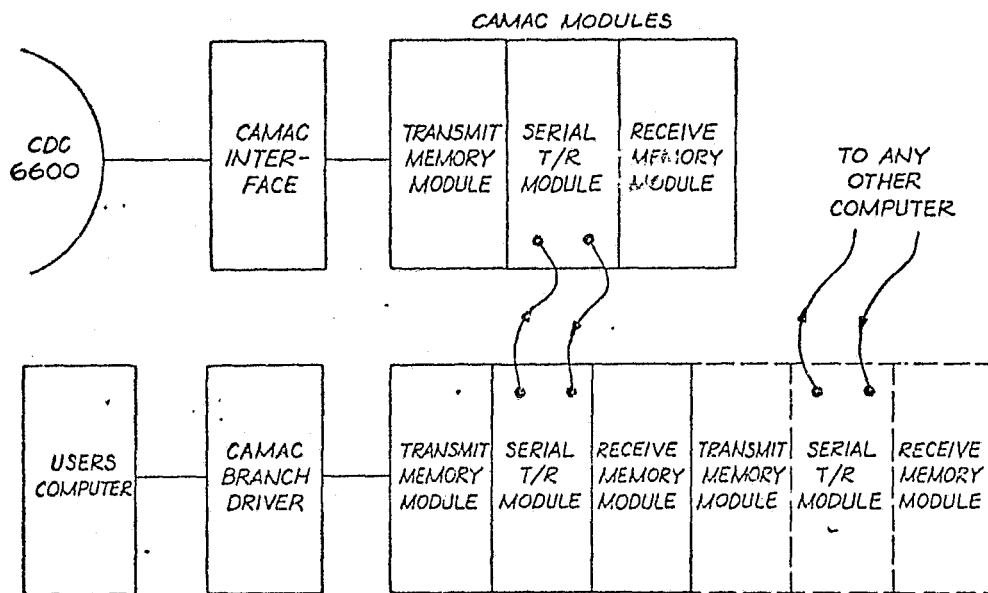


Figure 3-2. BISON Intercomputer Network

ORIGINAL PAGE IS
OF POOR QUALITY

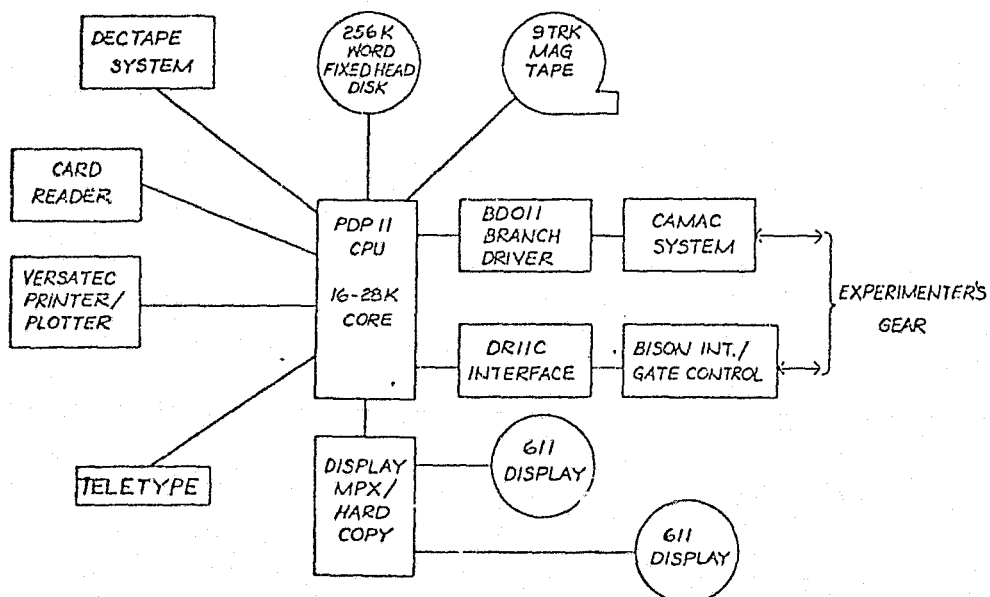


Figure 3-3. A Typical BISON PDP-11 Configuration

Operating Systems - As mentioned above, a number of operating systems are available to be used at Fermilab. The PDP-11 operating systems include DEC's Real-Time RT-11 disc operating system, DEC's RSX multitask system, and Fermilab-developed systems SPEX and BSX. SPEX is the spectrometer executive developed by Experiment 96 for use at the Meson Laboratory's Single-Arm Spectrometer. It is a core-resident multitask supervisor that swaps tasks in from the disc as they are required and frees core for additional swapping as tasks exit.

BSX is the primary operating system for PDP-11's. It is a core-resident real-time interrupt-driven multitask supervisor that is operated with DEC's DOS. The PDP-11's trap directives are used to control the various Tasks, by changing the Program Counter (PC) and the Processor Status Word(PS). A trap is effectively an interrupt generated by software. When a trap occurs, the contents of the current PC and PS are pushed onto the processor stack and they are replaced by the contents of a two-word trap vector containing a new PC and new PS.

Application Programs - Appropriate subroutines are provided at Fermilab to allow FORTRAN application programs to operate in real-time. In addition, FORTRAN callable utility subroutines are provided to retrieve parts of words, shift bits of words, modify words, and work with specific addresses.

The application programs used at Fermilab are numerous and constantly changing, as contrasted to those at DAPCS where a few programs are used month after month. At Fermilab there are two ways to implement CAMAC systems. One used the Kinetic Systems serial branch driver in an inexpensive data transfer approach. For the Kinetic Systems KS0011, a table of directives, properly formatted, provide experiment control and data transfer under program control (i.e., no hardware block transfers). The FORTRAN callable routine, KS0011, transfers the desired CAMAC commands.

The other method at Fermilab for data transfer, including block transfer of data by Direct Memory Access (DMA), uses an EG&G BD011 branch driver and at least one DEC Device Register Interface (DR11-A). The BSX operating system provides for much of the task priorities, interrupt, and CAMAC handling. For example, a PDP-11 assembly language application program can define eight word task tables, define specific tasks, and proceed to accomplish the tasks under program control solely by BSX. The trap directives are all FORTRAN callable.

While BSX can perform CAMAC handling, the general FORTRAN approach uses BSX through various subroutines. A labeled common (CAMCOM) of seven one-word integer variables is used to provide for "CAMAC BRANCH DEMAND interrupts," status of buffers, etc.

The routine CAMVL is used to initialize the Branch Driver (BD011) and up to four DR11-A's. This allows up to four high-priority (DMA) events; i.e., non-BD011 interrupts, to be initiated by DR11-A's. Included in the Call statement are the interrupt vector address and respective list of CAMAC commands for up to four interrupts. Also included is a word controlling the mode in which event-associated CAMAC processing will be performed. The lists of commands are coded with the first five words defining event variable, maximum word count, release and initialization flags, and a "non-interrupt routine" address before the actual CAMAC commands which are regular BD011 crate selection-word count and instruction words.

In addition to the above DR11-A interrupts set up by CAMVL and normal Branch Demand interrupts, explicit program calls to the routine CAMIO will initiate specified CAMAC operations. This provides for lower priority CAMAC operations. Up to eleven calls may be queued up at a given time and these tasks specified by task vectors will then be processed on a FIFO (first in first out) basis.

Other FORTRAN calls are available to handle buffers, branch demands, errors, etc. and also to enable or disable the DR11-A interrupts. These, along with the FORTRAN utility programs, must have the task vector format to conform to the standard BSX task table.

CAMAC Driver - The system does not have a directly identifiable CAMAC driver. The Operating System, BSX, in many respects serves the function of a CAMAC driver; i.e., CAMCOM (common block), CAMVL and CAMIO combine to establish the CAMAC command that BSX then passes to the branch driver (BD011) and, hence, to the CAMAC system. Similarly, a call to KS0011 passes the desired command to the KS0011 branch driver. The call itself includes the list of commands coded per simple setup and action words. A standard FNA is in the action word while the crate number, number of CAMAC words to be transferred, and the control bits are contained in the setup word. This, within the established formats, both parallel and serial CAMAC branch drivers are FORTRAN programmable.

3.2.3 Los Alamos Scientific Laboratory

3.2.3.1 General Description

The software system for data acquisition in experiments using CAMAC at Los Alamos is designed to support high-speed data acquisition on CAMAC systems controlled by a PDP-11 via a microprogrammable branch driver (MBD). The software system provides CAMAC drivers for the PDP-11/branch driver combination, utility routines and a special task-oriented language interpreter to facilitate user application program development.

The arrangement relative to the PDP-11 unibus is illustrated as Figure 3-4. The dotted arrow indicates the additional six CAMAC crates that could be added to the system. The event trigger is a special CAMAC module that was designed to identify the various different classes of events, to facilitate control and testing of the equipment, and to facilitate communication between different modules in the system.

The MBD contains a full-fledged processor with an instruction time of 350 nsec, eight priority structured DMA channels which share access to the CAMAC branch and unibus, and a sharable control memory (1024 words). The MBD controls the CAMAC branch, is capable of performing DMA data transfers to PDP-11 memory, and can interrupt the PDP-11.

The MBD was required at Los Alamos because many of the experiments have high event rates and very high data rates. Their minimum system requirements are: two DMA channels for experimental data, one DMA channel to display accumulated data, and one DMA channel for communication with Los Alamos Meson Physics Facility terminal computers.

The three major parts of the MBD are: the PDP-11 computer interface, the CAMAC branch driver, and the microprocessor. The computer interface has five 16-bit registers. These are: 1) memory address register (MAR), 2) memory data register (MDR), 3) control and status register, 4) program data register, and 5) mask register. The MAR and MDR are DMA channel registers controlled by the processor which controls the PDP-11 data during all DMA transfers.

The branch driver is a conventional design with three basic registers; the 16-bit command register (CNAF), the 24-bit branch data register, and the 24-bit graded-L register. In order to get around the problem of the

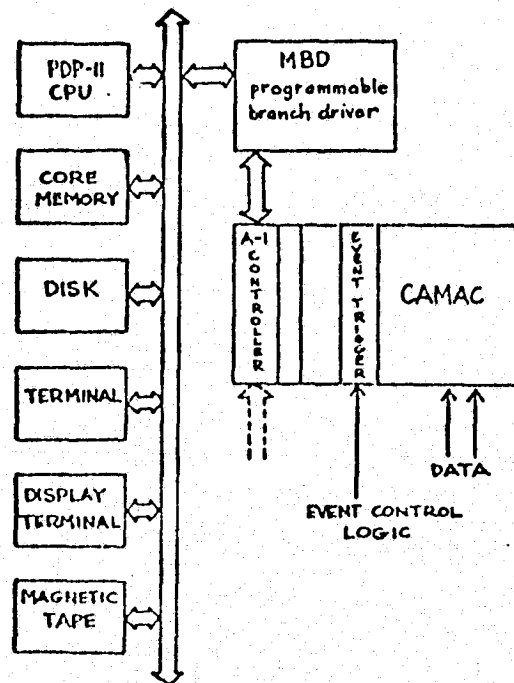


Figure 3-4. Los Alamos Scientific Laboratory PDP-11/CAMAC Data Acquisition System

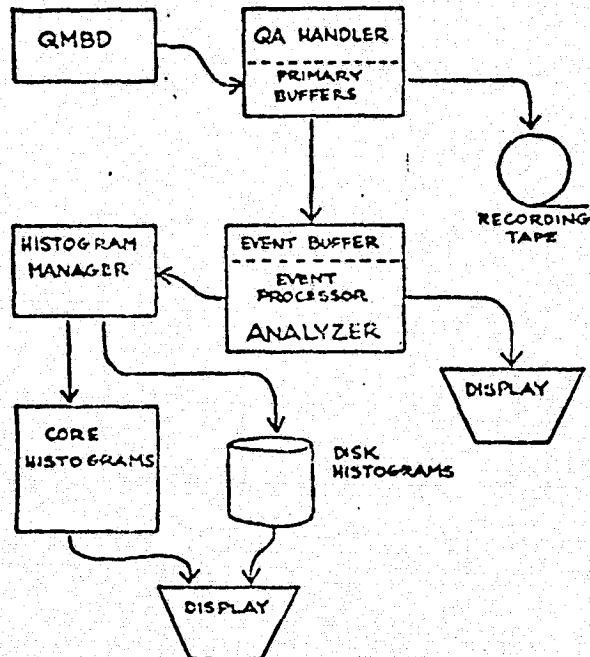


Figure 3-5. Los Alamos Scientific Laboratory Software System Diagram

command requiring seventeen bits, three different command types are defined: read, write, and control/test. Bit F8 is omitted from the command word and is provided by the processor. The processor is in complete control of the branch driver.

The microprocessor is the control device that gives the MBD the speed and flexibility such that transfers between registers is faster and requires less hardware than with gates alone. The arithmetic and logic unit (ALU) is the heart of the processor and it connects the Source bus and the Destination bus. One microinstruction of the ALU transfers words between any of the registers connected to the buses. An important function of the processor is to multiplex and control the eight DMA channels. In addition, the processor controls all communication between the CAMAC branch and the computer I/O. This frees the PDP-11 for use in real-time data computations.

The Event Trigger Module provides for 32 external signals to be entered into the system via the CAMAC dataway on a priority basis. Receipt of an external trigger generates a Look-at-Me, causing the MBD data acquisition channel to begin execution, and it signifies that one of the 32 user-defined events has occurred. The trigger module also provides "busy" outputs that can be monitored.

3.2.3.2 Software System Description

Operating System - The PDP-11 operating systems at Los Alamos are DEC RSX-11D or RSX-11M, interrupt-driven multitask supervisor systems. Where the Fermi-lab operating system (BSX) accomplished most of the CAMAC operations through task tables, at Los Alamos most of the CAMAC data acquisition is accomplished by the application program and CAMAC (QA) handler. Figure 3-5 is a simplified software block diagram. The operating system is involved with the tape unit, the display, disc histograms, core histograms, and the histogram display; i.e., the computer non-CAMAC I/O functions. A portion of the QMBD, which is a resident data-acquisition MBD code, can also be considered part of the operating system.

Application Programs - At Los Alamos, the application programs which must be prepared specifically for each given application, and, hence, supplied by the user of the programs, are the event descriptions, the event processors, and the initialization sequence. The event descriptions are provided via a special

easy-to-use event specification language (Q system), while the event processors and the initialization can be written as FORTRAN subroutines.

An application program tailored for each experiment is referred to as an analyzer program and is written, translated, and task-built with a standard structure. The end result is two files: one an object file containing the MBD code which acquires the data, and the other an RSX-11 task that processes the data acquired. Each analyzer program is given a name and is used elsewhere when referring to the particular analyzer. Each device is given a name and is unique by including a module name (e.g., KS3610, Kinetic Systems 3610 scaler) and its address C, N, A. The module names must be part of the Q system which defines the legal CAMAC operations on the modules. Each event for the analyzer program is given a number, 0-23₈, with 24-32₈ reserved for special system functions. Operand commands (e.g., RD24, read 24-bit data) may be defined for multiple devices which were previously defined. Control and test commands may be specified. General CAMAC FCNA commands can be specified. The Q system translates the above and other normally-required CAMAC functions to produce the required data acquisition and data analysis files. The MBD does the data acquisition and the PDP-11 does the data analysis.

What the above describes is a coded procedure to simplify the preparation of the application program. The specific steps or statements are defined in Los Alamos documents and must be followed with a certain rigor in order to satisfy the standard established format. The result is an efficient coupling of data acquisition using the microprogrammed branch driver (MBD) and data analysis using the PDP-11.

CAMAC Driver - As implied in the preceding section on application programs, the Q system provides the required CAMAC driver. That is, the Q system at Los Alamos interprets the input command statements and develops the required code for the MBD to pass the desired commands to the proper CAMAC modules

3.2.4 Aluminum Company of America (ALCOA)

3.2.4.1 General Description

The CAMAC support library for industrial systems at ALCOA provides an extensive library of computer-independent software modules to facilitate the development of diverse portable applications programs in standard ANSI FORTRAN

supplemented by standard ISA bit manipulation routines. A wide repertoire of computer/branch-driver-specific CAMAC drivers are available with a standard FORTRAN call sequence. A flexible logical device table generator scheme is included to handle diverse or variable hardware configurations with minimum software impact.

In order to make the software system applicable to a diverse range of industrial process control applications using many different control processors, a conscious effort has been made toward standardization; i.e., CAMAC hardware and ANSI X3.9-1966 FORTRAN software. Application programs are normally prepared in FORTRAN, and the hardware system support programs developed and cataloged by ALCOA are FORTRAN callable. In addition to the hardware system support programs; i.e., computer-CAMAC driver programs ALCOA has high-level (FORTRAN) test programs, adaptor programs, and general utility programs.

3.2.4.2 Software System Description

Operating Systems - Specific operating systems are not required, and any system with a standard ANSI FORTRAN compiler for a given computer can be used within the operational limitations of that system.

ALCOA has standardized their approach to CAMAC for various operating systems. This standard is based on use of four registers for interfacing (i.e., computer bus to CAMAC highway) which are used for the following: 1) COMMAND, 2) ADDRESS, 3) DATA, and 4) STATUS. These generic information classes are kept separate and intact. The COMMAND register is used right-justified when mapped from register or data paths of more than five bits such as the computer I/O bus. As for the ADDRESS register, the bits are given right-justified as C, N, A. Twelve bits are used for a parallel highway, while fifteen bits are required for a serial highway. Each crate must have a unique address in the software and overall system structure. This becomes an "effective" crate address in multiple-highway systems and the translation between effective address and physical crate address on a particular highway must be done in the highway or computer-port selection software. For the DATA register, which requires twenty-four bits on the CAMAC side of the interface, sign extension is used with right-justified data bits when the computer word exceeds twenty-four bits and two or more non-CAMAC registers must be cascaded with sign extension for computers with less than 24-bit words. So far as the

STATUS register is concerned, the following is the standard: a) the least significant bit indicates a No-Q response (i.e., it will be 1 if $Q = 0$), b) the next least significant bit indicates No-X response, c) the third least significant bit indicates a highway error, d) the next four bits are not to be used (reserved for future definition). A recommended option is that the eighth or higher bit be assigned as an "interface busy" bit. Then, when the software goes to read the status information, it can readily verify that the CAMAC operation has been completed. In most cases, the computer's sign-bit will be most useful for this purpose.

Application Programs - Specific CAMAC process control programs at ALCOA are usually high-level programs written in ANSI Standard FORTRAN X3.9-1966. These can be implemented by using a Logical Device Table; i.e., a table formatted to assign a logical device number (LDN) to each device in a system. Utility programs are available to generate the LDN's, store this array in COMMON, and to modify the table of LDN's as required. Each LDN can then be used in the application program as the argument in calls to specific functional handlers; e.g., INTEGER FUNCTION INCHINT (LDN) which returns an integer value from the respective plug-in module's Group-1 register. Other FORTRAN callable functions or subroutines are used to handle data words in arrays, singly, or in bits; to test LAM's; etc.

ALCOA has implemented the ISA-S61.1 Procedures (Instrument Society of America) as the standard method for FORTRAN manipulation of bit strings. Programs are provided for various computers when the manufacturer of the computer does not provide for the ISA procedures.

CAMAC Driver - There are many CAMAC drivers used at ALCOA: i.e., one for each computer-branch driver combination used. They are all used the same with simply CALL CAMAC (FUNCT, ADDR, DATA, STATUS). For multiple highway, the driver for each branch driver is renamed and CAMAC is then made as a highway selection code.

3.3 SPACELAB SOFTWARE SYSTEM

The primary hardware interface for data acquisition and control between Spacelab and investigator-supplied experiment instrumentation is the Control and Data Management Subsystem (CDMS). This subsystem incorporates a Mitra 125S general purpose computer that is dedicated to support of experiment operations. A complete set of software, including operating system and user program development bits will be available for that computer. An overall description of the Spacelab software system from the user's standpoint is given in Section 4.5, Software, of the "Spacelab Payload Accommodation Handbook," ESA, May 1976. This material is reproduced in Appendix II for this volume. More detailed descriptions of the Spacelab software system and the CDMS operating system are contained in "Software Specification," SR-ER-0001, ERNO, August 1975; and "CDMS Operating System, Package Design Specification," SS-ER-0012, ERNO, September 1975.

The software provided for users with the CDMS system covers all aspects of software development, integration, testing and operation, including in-flight command and data handling. A functional breakdown of the total Spacelab software system is shown in Figure 3-6.

The investigator's experiment specific software or application program will interface with the experiment computer operating system (ECOS) and can make use of certain facility-type software incorporated in the flight application software packages (FLAP).

The ECOS will be core-resident except for display routines and will consume approximately five percent of the computer execution time. It will handle the scheduling of tasks and allocation of resources with executive routines for task scheduling, memory management, time management, computer resources management, and asynchronous task handling.

In addition, it will provide the software interfaces (input/output drivers) to the remote acquisition units (the hardware interface between the CDMS and experiment instrumentation), the CRT display units and the operator keyboards, as well as the standard peripherals. The primary use of FLAP for experiment operation will be in obtaining Spacelab subsystem operating information such as resource availability.

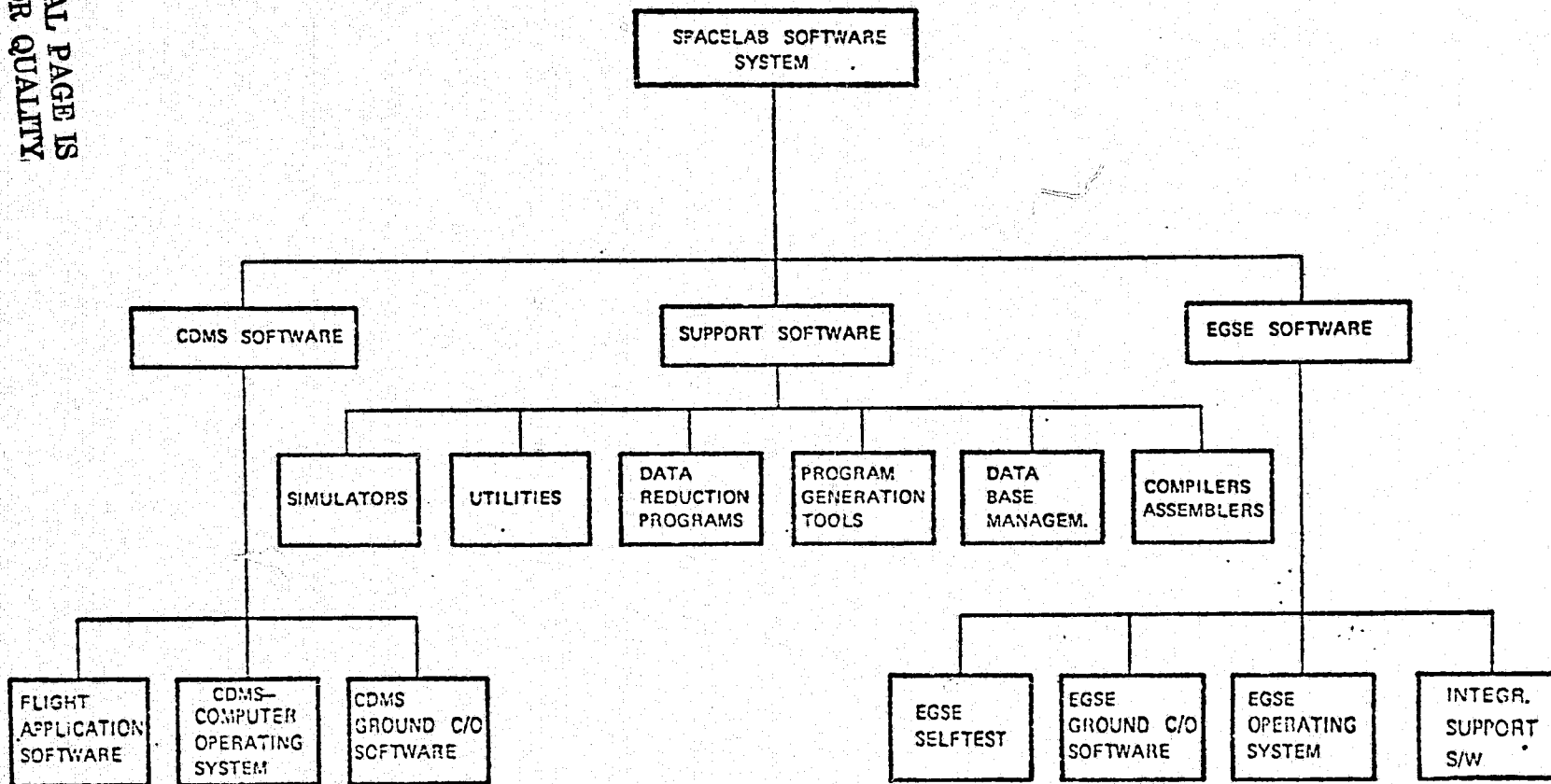


Figure 3-6. Spacelab Software System Functional Breakdown

The software development aids are designed for operation on an IBM 370-series host computer. There are two HAL/S language compilers, one producing IBM 370 machine language and one producing Mitra 125S machine language. Programs in 125S language can be executed on a 370 with the use of a 125S simulator supplied as one of the development aids. There will also be two version each of a 125S macroassembler and an editor. One version will run on a 370 and the other will run on a 125S. A simulator running on the 370 for the entire CDMS exclusive of the 125S will be available for program testing in conjunction with the 125S simulator.

Using the above set of development aids, the investigator will be required to write his experiment-specific programs in either HAL/S or in Mitra 125S assembly language. Unfortunately, in spite of the completeness of the available software in terms of types of functions implemented, this is a serious deficiency from the typical user's point of view. No provision is made for users to write programs in any of the high-level software languages commonly used for laboratory data systems. Compounding this problem for the investigator who is willing to work in assembly language is the selection of a computer with which the typical user is totally unfamiliar. The user will thus be forced to invest in the necessary time for his programmers to become familiar with these languages. In addition, as is always the case when a new language is first learned, the efficiency of the programming effort will be low initially and the execution efficiency of the resulting code will also be less than optimum for early efforts.

These potential problems will be somewhat alleviated if NASA is able to supply standard software modules to the investigators to handle routine repetitive operations such as accessing CAMAC hardware modules. As a result of these factors, it will be very important for NASA to supply as much standard software as possible to somewhat relieve the burden placed on individual investigators.

3.4 CAMAC SUPPORT SOFTWARE FOR SPACELAB

In investigating four existing approaches for software to support the use of CAMAC hardware, it was found that several different concepts were used, sometimes even to meet similar requirements. The Spacelab experiment environment does not exactly match any of these four approaches and, in fact, includes some aspects of each. In considering the software requirements of the payloads analyzed in Task 2, a natural division into two distinct categories was found. The first of these was the use of CAMAC to implement facility-type functions and the second was the use of CAMAC to implement experiment-specific instrumentation.

3.4.1 Software for Facility Use of CAMAC

The simplest of these two cases, in terms of selection of the best approach, is the facility use of CAMAC. In this case, the requirements change either very little or not at all as a function of time. The facility is reflown many times and accommodates new instruments from time to time but, in general, it supplies the same support functions on all flights. Because the CAMAC software is written only once and used for many flights, a somewhat higher programming cost can be justified in order to achieve more efficient software performance in terms of resource requirements: execution speed and memory size. It is universally accepted that programming in assembly language can produce more efficient codes although at higher cost than if a higher-level language is used. Because of the nature of the long-term utilization of the CAMAC implemented facility-type equipment for Spacelab, the use of assembly language programs specifically tailored for each application is the best choice. Responsibility for preparing the programs will usually best be left with the facility hardware development organization.

3.4.2 Software for Experiment Use of CAMAC

For experiment-specific use of CAMAC, the variability of the software requirements means that software development will be a continuing process. Therefore, a convenient, user-oriented software system is preferable in spite of its reduced operating efficiency. In this case, a wider variety of techniques appears to be applicable, and the tradeoff among alternative approaches is not always clear. This uncertainty is exemplified by the choice of different approaches to solve similar problems in the four existing

software installations surveyed. In the case of Spacelab, however, there is one major consideration that clearly drives the choice. Because of the unfamiliarity of the typical user with the programming languages used with the Spacelab computer, the burden on the user can be greatly relieved if NASA provides a set of standard software modules that can be called in HAL/S for use with the CAMAC hardware. The resulting programs will not be as efficient in utilization of resources as the specially-coded facility programs discussed above, but the cost of preparing new experiment-unique software for each mission will be greatly reduced.

The optimum system for reducing programming effort uses a hierarchy of several routines. The lowest level is a single CAMAC driver that provides the basic hardware/software interface. This handler is very hardware specific, depending on the computer and the CAMAC branch driver combination being used. It would be written in assembly language for the Mitra 125S and provided as part of the ECOS and the simulators for the 370. If a standard branch driver is developed for use with the Spacelab CDMS because of the high degree of commonality expected for this system-common hardware element (see Section 3.1.1, Volume II), the development of a driver for the 125S/branch driver combination would be a one-time effort that should be done in conjunction with the branch driver hardware development.

The next level of subroutines would be the standard CAMAC software modules. These software modules would exist in a one-to-one correspondence with each type of CAMAC hardware module. They would be callable from HAL/S and, in turn, would call the CAMAC driver. They would allow reference to individual hardware units by means of logical unit number rather than physical locations within the CAMAC crate system. In this way, the user software would be independent of the specific hardware configuration. The correspondence between logical unit numbers and hardware location would be established by an initializing routine prepared by the integrating contractor. This approach is best illustrated by the logical unit table generation scheme included in the ALCOA CAMAC support library. It would minimize the amount of software modification required when the hardware configuration is changed.

The standard CAMAC software modules would be written in assembly language since they will be unchanging with time. In order to assure the automatic compatibility of CAMAC hardware modules with the software system, the

standard CAMAC software module corresponding to each hardware module should be developed in conjunction with the hardware.

The highest level of software would be the unique user-supplied application program. This would be written in the high-level language, HAL/S for each experiment and would access the CAMAC hardware by calls to the standard CAMAC software modules. The user program would generally consist of a main observation program along with at least two subroutines, one for instrument control and one for data acquisition.

The CAMAC handler and standard software module would be provided to the individual experimenter by NASA. Either the experimenter or his instrument contractor would prepare the experiment-unique software in each case. Although the resulting total software is not optimized in terms of execution time or memory size, it certainly provides the lowest risk development in terms of schedule and cost.

The software cost impact of the use of CAMAC will depend on the host software environment which is available. If the complete Spacelab software system is available, the addition of the standard CAMAC software will greatly simplify the user effort devoted to input/output data transfers between the Spacelab computer and his experiment hardware, but this represents only a portion of the user's application program development task. The main advantage to the user would be the capability to write his applications program in direct correspondence with the hardware with which he is most familiar - the CAMAC modules used in his experiment rather than the Spacelab CDMS remote acquisition units. If convenient, user-oriented, input/output drivers for the Spacelab RAU are available in the Spacelab software system, the cost differences due to the use of CAMAC equipment will probably be slight. On the other hand, if the available Spacelab software support is limited or inconvenient to use, the availability of standard CAMAC software can save a considerable amount of the user effort that would be required to develop special input/output drivers specifically for his experiment.

3.5 SOFTWARE REQUIREMENTS FOR REPRESENTATIVE PAYLOADS

The Shuttle Infrared Telescope Facility (SIRTF) and the X-ray/Gamma-ray payload were selected as specific examples to illustrate the software implementation for payloads using CAMAC equipment.

3.5.1 Shuttle Infrared Telescope Facility Software

3.5.1.1 Major Categories of SIRTF Software

In considering what is needed to operate SIRTF during a Shuttle flight, it is evident that two major categories of software functions are required. The first is that which monitors and controls the operation of the telescope facility, itself, and the other is concerned with data processing and control of whatever focal plane instruments are being used in the observations being performed. The software in each of these two categories operates mostly independent of that in the other category and the treatment of each category as far as the creation and configuration management of the programs will be different.

A wide assortment of programs and subroutines will be required in the category of facility software. An automatic initialization routine will be necessary to activate the telescope after the Shuttle has been launched and has arrived on station. This will include operations for uncaging and starting cryogen flow. In association with this initialization routine, another program will be required to perform an automatic checkout and setup of the telescope including such functions as focusing and establishing optical alignment using a laser source. This latter function is especially critical because the telescope will be incapable of adequate alignment until it experiences zero-g environment. Additionally, there must be a facility program to handle the overall pointing and control of the telescope during operation. This program must interact with the Orbiter attitude control programs and with the observation programs for the individual experiments. Finally, an automatic shutdown program will be required that cages the telescope and safes the facility in preparation for landing.

The software associated with each of the individual experiments will be primarily concerned with providing operational control to the instruments and processing the data from the infrared observations. The routine that governs the overall data gathering operation must interact both with the

data processing program for the instrument and with the facility telescope pointing programs. Additionally, the experiment software must provide for the display of experiment data on the Spacelab CRT's and also be able to interpret inputs from the Spacelab keyboard.

In order to determine the applicability of modularized CAMAC software to SIRTf, consideration must be given to the different characteristics of the two categories of software.

Specifically, since the software associated with the facility is independent of whatever instruments might be mounted in the focal plane, it will be developed as a part of the development of the facility on a one-time, non-recurring basis. The software associated with the focal plane instruments, however, will change with each new instrument and will continue to be modified as the experiments are modified. These considerations indicate that the facility software is better developed uniquely for each of the facility requirements and that it is only in the case of the experiment software that benefit might be had from the modularized concept.

Additionally, it should be pointed out that the modular CAMAC concept in the software presupposes the existence of CAMAC hardware with which to interface. In Section 3.3 of Volume II, it was seen that except for the housekeeping and fine pointing functions, all of the application of CAMAC hardware was to the signals from the focal plane instruments. There was very little applicability for CAMAC in the rest of the facility.

For these reasons, the analysis that follows will concern itself with the application of CAMAC modular software to the instrument-related category of SIRTf software almost exclusively. The only exception will be that the facility housekeeping and fine pointing functions will also be considered.

3.5.1.2 SIRTf Software Requirements and System Design

The specific software functions required to support the operations of the five SIRTf instruments are analyzed below. Also included is an analysis and discussion of what is required to operate the housekeeping and fine pointing systems if they were implemented with CAMAC hardware as discussed in Task 2 of this study.

Filter Photometer - Operation - When the filter photometer instrument is flown, it will have a preplanned program of observation to execute. This program will be established based on prior knowledge of Shuttle orientation, sun-moon-earth positions, and the requirements of other instruments during that mission. The observation of a single object (target) will proceed as follows:

The payload specialist will tell the experiment computer (via keyboard input) that he is ready to move to the next object in the filter photometer observing program. The computer retrieves the coordinates of that object from the mass memory and from the Orbiter computer, gets the current orientation of the Shuttle. From this it computes the operations necessary to acquire the target and display it for the payload specialist. After the target area has been acquired, the computer retrieves from mass memory and displays for the payload specialist instructions about how to observe the specific object. This will include finding charts (to aid in fine pointing acquisition), expected signal levels, filters and apertures to be used, and any special instructions for carrying out the observation.

The payload specialist will then perform the fine pointing functions necessary to acquire the particular object to be observed within the target area. The detailed operations required for this acquisition will depend on how the facility ultimately provides for fine pointing control (e.g., joystick, keyboard input, pushbutton).

Having acquired the object, the payload specialist will call up the observation mode program in the experiment computer and input the parameters necessary to perform this observation. These parameters will include specifying the amount of time to observe the object before moving to a nearby piece of background sky and the number of times to repeat the object-sky-object cycle. He will also specify the sequence of filters and apertures to be used in making the observation. In one mode of observation, he may require that a particular signal-to-noise ratio be attained before moving to the next filter and aperture selection.

Having made these inputs, he will initiate the observation process that will then proceed under computer control. As the observation is made, the experiment computer will perform all routine monitoring of instrument status parameters and control the fine pointing of the telescope. The computer

will also perform real-time processing and display of the infrared data and, via its interface to the Orbiter, output this data to the telemetry system.

Filter Photometer - Software Requirements - The major software functions required to operate the filter photometer as just described are shown in the top level system diagram of Figure 3-7. The Spacelab computer system is indicated on the left of the figure and the CAMAC hardware used to support the facility and the filter photometer is shown on the right. Each of the blocks in between represents a module of software operating within the experiment computer. The diagram does not show the separate subroutine that is used for the initial retrieval of information from mass memory to give the payload specialist the background information required for each observation.

The central control of any given observation resides in the observation mode program. This program receives the keyboard instructions of the payload specialist and uses them to execute the details of the observation. In order to do so, it must interface with the facility subroutines for fine pointing and housekeeping so that it can control the telescope. It also must interact with the subroutines that process the data from the photometer since it reacts to the signal-to-noise ratio of the infrared detector and to other of the instrument parameters in determining the amount of time to be spent on any given measurement.

The data acquisition subroutine is specifically oriented toward the filter photometer. It contains all of the data reduction parameters and computation algorithms necessary to process the signal from the infrared sensor. By accessing the CAMAC module subroutines, it gets the digitized data from the sensor. Information about filter position, aperture and photometer temperatures are obtained from the instrument control subroutine. Pointing and frequency of spatial chopping data come to it through the observation mode program. Upon request it provides processed data to the payload specialist.

The instrument control subroutine monitors the status of the photometer and generates control instructions for the selection of filters, choice of apertures, and the insertion of the blackbody calibration source into the beam. It accesses the CAMAC module level subroutines as required to

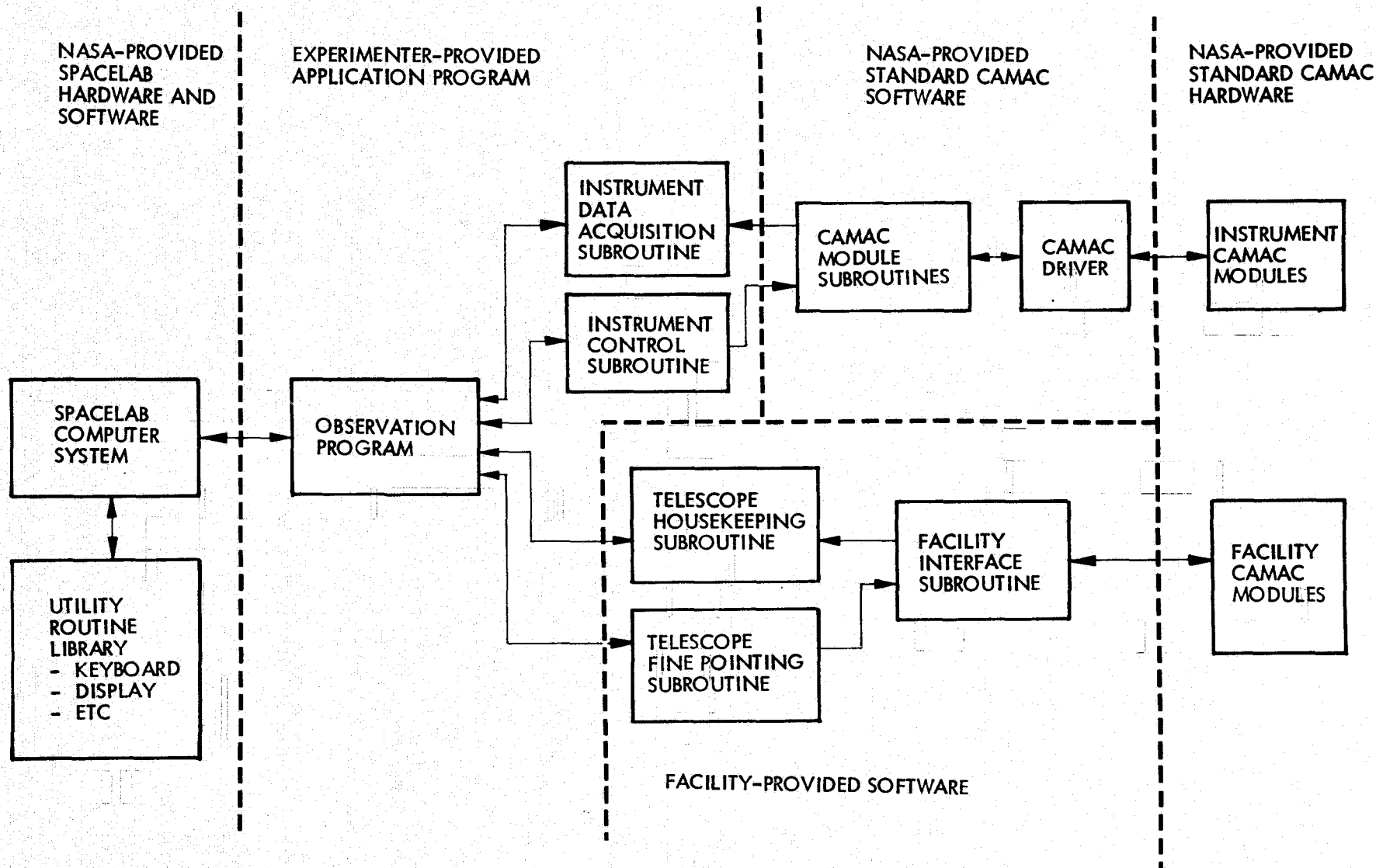


Figure 3-7. SIRTF Software System Diagram

retrieve data from the photometer and generate control signals. Upon request by the observation mode program, this subroutine provides the instrument status parameters for display or changes the instrument configuration for the next stage in the measurement process.

Both the data acquisition subroutine and the instrument control subroutine communicate with the photometer through the CAMAC module level subroutines. There is one of these for each module in the CAMAC crate supporting the photometer functions. They, in turn, access the hardware through a CAMAC driver that is used in common by all of the CAMAC module software. It is this driver that forms the software side of the software-hardware interface in the CAMAC system.

The display and keyboard subroutines shown in the figure handle the details of operating those two pieces of hardware. If CAMAC is used in these systems, then they also would be accessing module level CAMAC subroutines.

The fine pointing and housekeeping subroutines interact with observation mode program for control of the telescope facility. They will be discussed separately in a later section.

Filter Wedge Spectrometer - As discussed in the earlier analysis of this instrument, it is essentially identical operationally to the filter photometer. It would require identical software and could be operated in the same fashion except that measurements at many more filter positions would be required.

Grating Spectrometer - The primary difference between this instrument and the filter photometer is that here the observation mode program and the instrument control subroutine must use and control the grating in the instrument. The particular spectral settings used in a given measurement will vary depending on the nature of the object and the purpose of the measurement. The choice of these settings will be input to the observation mode program by the payload specialist at the beginning of each observation. The data acquisition subroutine will have to incorporate the grating orientation into its interpretation of the signal from the sensor. Otherwise the grating spectrometer requires the same software and operates in the same fashion as the filter photometer.

Detector Array - As discussed in an earlier section, the detector array is functionally identical to the filter photometer except that it generates 256 identical signals and allows simultaneous measurement and mapping of many points in a region of interest. Each of the 256 signals may be thought of as that from a single filter photometer and is processed in an identical fashion. However, the capabilities of the data processing subroutine must be much expanded to handle all of these signals at one time. It also must average the signal-to-noise calculation over the entire array of signals in order to present reasonable signal-to-noise information to the observation mode program. The display software for this instrument should be capable of presenting the infrared map of the target area as it is accumulated by the sensors. The only difference in the instrument status subroutine from that for the filter photometer is that control signals must be provided to operate the pulse generator which clocks signals out of the detector array.

Fourier Spectrometer - The software system required to operate the Fourier spectrometer is organized the same functionally as that shown for the filter photometer. However, the operations performed in the observation mode program, data processing subroutine and instrument status subroutine and the uses made of the fine pointing are significantly different from those in the filter photometer system. The primary reason for these differences is that in this instrument the mechanical operation which causes the signal variation detected by the infrared sensor is the motion of the Michelson mirror in the instrument itself and not the spatial chopping with the second folding flat of the telescope.

One major result of this difference is that in this system the instrument control subroutine must be more sophisticated than a photometer-type system. It must precisely control the motions of the Michelson mirror and must continually keep the data processing subroutine informed of these motions. In turn, the latter subroutine will have to precisely coordinate its samplings of the sensor output with the movements of the mirror. A sizable data storage array will be required in which to accumulate the samples from each of the different mirror velocities. In order to present the data as a spectral distribution, the data acquisition subroutine will either have to contain a fast Fourier transform algorithm or be able to call upon one in the Spacelab software system.

The observation mode program for the Fourier spectrometer will maintain overall control of the measurements but will not be as directly involved in the measurement process as it was in the photometer system. Once it has put the object in the spectrometer aperture by controlling the fine pointing system, it essentially turns control over to the instrument control and data acquisition subroutines until the measurement is complete. At that point, it resumes control and may call for a display of the spectrum or may repoint the telescope at a nearby section of sky to take a background measurement.

Housekeeping and Fine Pointing - The software to monitor the facility housekeeping signals and to control the fine pointing of the telescope will operate within the experiment computer and interface into CAMAC equipment but because it is a permanent part of the facility and independent of whatever focal plane instruments are flown, it will be structured somewhat differently from the CAMAC software associated with the instruments. In order that they may be readily modified as the facility evolves and in order to be more easily adapted to the various observation mode programs, the fine pointing and housekeeping functions will be handled by separate subroutines. However, instead of calling CAMAC module level software to interface to the hardware, these subroutines will access a facility interface routine which itself directly accesses the hardware.

The facility interface routine is written in the assembly language of the computer. It is effectively an amalgamation of the CAMAC routines and the CAMAC driver. In the case of facility software, this amalgamation is acceptable because the hardware system is fixed and not continually being reconfigured as it is for the instruments. Because the facility interface routine accomplishes the software/hardware interface in a single assembly language stage, it is more efficient than the corresponding interface in the instrument software system.

The fine pointing subroutine itself performs all of the real-time functions necessary to close the control loop on the orientation of the second folding flat of the telescope. Through the interface routine, it relies heavily on the real-time interrupt capabilities of the computer to react to changes in the pointing status of the telescope as they are detected by the quadrant error sensor.

The housekeeping subroutine also depends on the real-time interrupt structure of the computer in that it must respond to any changes in status of the parameters it monitors. Additionally, it performs regular, routine checks of all signals in the housekeeping system and stores processed values for presentation to the observation mode program upon request.

3.5.1.3 CAMAC Module Software Requirements

The requirements for CAMAC module level subroutines can be derived in a straightforward fashion from the requirements for CAMAC hardware that were established in Task 2 of this study. Table 3-1 is a software requirements analog of Table 3-16, Volume II, which summarizes all of the CAMAC hardware requirements for SIRTf.

The one difference in the software table is that there is no need to indicate the number of module subroutines of a given type that are required for each instrument. At most, only one of each type will be required no matter how many modules of that type are used in the hardware system. This reflects the fact that a single CAMAC module level subroutine can be used to service many hardware modules. For each module, the subroutine is called with an argument that specifies which hardware module is to be serviced. The only modules for which separate subroutines may be required are the fast ADC's used in the detector array. This will depend on the frequency of sampling for that system which in turn depends on the frequency of spatial chopping that is used. It should also be noted that a separate type of subroutine is specified for the multichannel ADC's due to the slightly different software structure required by the added complexity of addressing a specific ADC channel.

Because the housekeeping and fine pointing systems access a custom-designed interface routine as discussed earlier, they are not listed in the table. Since they are the only subroutines that access digital-to-analog converter modules, no module level subroutines for this function are required.

Not shown in Table 3-1 is the CAMAC interface level subroutine referred to as the CAMAC driver in Figure 3-7. This assembly language program accomplishes the detailed interface of the software system to the hardware system. It is accessed by all of the module level subroutines and is, therefore, required in the software system of every instrument.

Table 3-1. SIRTf CAMAC Software Module Requirements

CAMAC Module Subroutine	SIRTf Instrument				
	Filter Photometer	Filter Wedge Spectrometer	Detector Array	Fourier Spectrometer	Grating Spectrometer
Input Register	•	•	•	•	•
Pulse Generator			•		
Output Register	•	•	•	•	•
Stepping Motor Controller	•	•	•	•	•
ADC's					
Single-channel, fast	•	•	•	•	
Multichannel, slow	•	•	•	•	•
Multiplexer				•	•

The CAMAC module level subroutines and the CAMAC driver will be furnished to each instrument as a part of the package including the CAMAC hardware modules required by that instrument. After selecting his hardware modules, the experimenter for each instrument will also receive a user's guide to the module level subroutine associated with each piece of hardware he has chosen. Reference to these user's guides will greatly simplify the generation of his data acquisition and instrument control subroutines.

Additionally, the housekeeping and fine pointing subroutines together with the CAMAC-oriented facility interface routine are available to each experimenter. This will allow him to accurately control the fine pointing of the telescope by high-order software operations. He will not require a detailed knowledge of the subtleties of the fine pointing control loop or the intricacies of the software-to-hardware interface for this control. Similarly, all of the housekeeping and status information from the facility will be readily accessible to him either by calling the subroutine or by having the status variables passed to him in common.

Finally, the software necessary to drive the I/O units (keyboard and CRT display) will be part of the software package available to each experimenter. As mentioned previously, these will include the appropriate CAMAC module level subroutines if CAMAC is the hardware standard used to implement them.

Thus, only three major elements of the software need be rewritten for each of the instruments. These are the observation mode program, data acquisition subroutine and instrument control subroutine. These can all be written in the high-order language of the experiment computer and all of their interface to the hardware will consist of simple calls to the appropriate subroutines. The use of CAMAC and facility software thus minimizes the software effort required as different instruments are developed for use in SIRTf.

3.5.2 X-Ray/Gamma-Ray Pallet Payload

3.5.2.1 Major Categories of X-Ray/Gamma-Ray Pallet Software

The X-Ray/Gamma Ray Pallet is composed of three independent instruments primarily intended for preprogrammed automatic operation. They require no specialized facility support from the Spacelab beyond the normal resource provisions and pointing capability. Because of these factors, no facility software is required to support the operation of these instruments other than the standard operating system for the Spacelab experiment computer. Each instrument does, however, require a unique set of experimenter-provided software which can readily be partitioned in the manner recommended in Section 3.4. The programming effort required to develop this unique set of software would be greatly reduced if the recommended set of standard software modules were supplied to the experimenter by NASA.

3.5.2.2 X-Ray/Gamma-Ray Pallet Software Requirements

Because of the similarity of the software requirements for the three instruments that form the X-ray/Gamma-ray pallet, Figure 3-8 can be used to represent the software and hardware interrelationships for each instrument. As shown in Section 3 of Volume II, all electronic requirements for these instruments can be satisfied with NIM and CAMAC hardware with the exception of a limited number of amplification and power supply requirements. In particular, all data and control interface functions are implemented with CAMAC modules and this interface is represented by the block at the far right side of Figure 3-8. The software interface is provided by a standard NASA-provided CAMAC driver. In addition, a set of standard, NASA-provided, CAMAC module level subroutines are used to reduce the programming effort for the instruments. These subroutines provide access to the individual

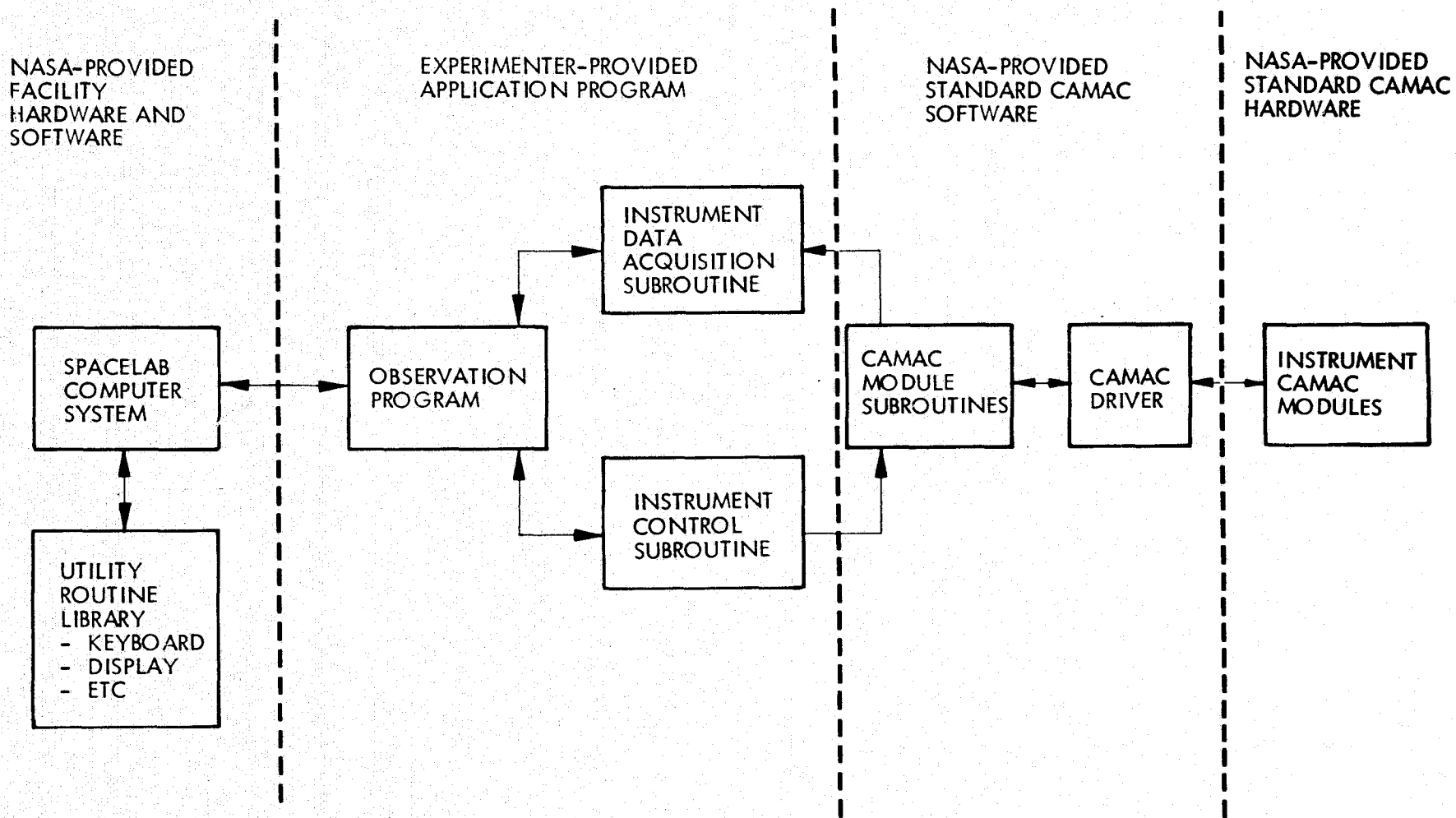


Figure 3-8. X-Ray/Gamma-Ray Pallet Payload Software System Diagram

CAMAC hardware modules from a high-level programming language on a one-for-one basis.

The only unique software that the experimenter must provide for an individual instrument is the main observation program and its associated instrument control and data acquisition subroutines. This set of unique software is executed in the environment of the Spacelab experiment computer system hardware and software and calls the standard CAMAC software modules as required.

Large Area Proportional Counter Array - The Large Area Proportional Counter Array has very few control requirements. In the normal data acquisition mode, the instrument is completely passive, responding on an event-by-event basis to detected X-rays. Attitude control operations, including pointing at sources and scanning regions of the sky with the optional modulation collimators, are provided by the normal Spacelab facility capability.

The instrument control subroutine satisfies two housekeeping-type functions. The ongoing activity is the maintenance of the gas pressure in the MWPC's within pre-established limits. This requires periodically measuring the pressure in each of the ten MWPC's by means of the transducers connected to CAMAC ADC's. When the pressure falls below the desired range for a given MWPC, the gas supply valve is actuated by a CAMAC output driver until the pressure reaches the upper limit of the desired range. The calibration function is an infrequent operation occurring at preprogrammed times. CAMAC stepping motor drivers are used to position radioactive sources in front of each MWPC and CAMAC position encoders are used to determine the exact source position in each case. These sources are left in front of the MWPC's for a fixed length of time and then retracted so that normal data taking can resume. The instrument control subroutine would utilize four standard CAMAC software modules, one corresponding to each type of CAMAC hardware module mentioned above, in addition to the standard CAMAC handler.

The data acquisition subroutine can react to the occurrence of an event in one of two ways, depending on how the event trigger is implemented in the hardware. If the instrument is assigned to a computer interrupt, event data can be acquired on a prioritized demand basis. In this case, the data acquisition software must be prepared in the form of an interrupt subroutine with appropriate entry, exit and register save functions as required

by the operating system. If the instrument uses a CAMAC input register for setting an event flag, then that CAMAC module must be periodically polled at a rate that is significantly higher than the expected event rate in order to avoid substantial counting rate type losses. The polling function can probably best be performed at the system level rather than by periodically calling a user-supplied program. In this case, the data acquisition subroutine could be activated by a software generated interrupt or in a less direct context change, by the time sharing system.

Once activated, the data acquisition subroutine would access the CAMAC hardware modules to recover the event specific data and the updated scaler values not directly associated with the occurrence of an event. This data would be buffered in the memory for logging to permanent storage and also made available to the instrument observation program. After obtaining the event data, the data acquisition subroutine would perform any necessary resetting and clearing of data buffers in the CAMAC hardware to prepare the instrument for the occurrence of the next event. A total of four types of standard CAMAC software modules are required to perform these data acquisition functions.

The observation program would coordinate the activities of the instrument control and data acquisition subroutines, in particular during calibration sequences. It would also provide for any on-line preparation of the data for quick-look displays used to verify proper operation of the instrument.

Bragg Crystal Spectrometer - From the standpoint of the software, the Bragg Crystal Spectrometer can be treated as two separate instruments, with the low-energy and high-energy spectrometers operated independently. In the normal data acquisition mode, both spectrometers respond to detected X-rays on an event-by-event basis. The instrument control subroutine must provide active control of the crystal position during data acquisition. The crystal is stepped through a preprogrammed series of positions to provide energy spectrum information and this observation cycle is repeated periodically. Attitude control operations are carried out by the normal Spacelab facility capability and are coordinated with the crystal observation cycling.

In addition to controlling the crystal positions during experiment observations, the instrument control subroutine also provides for gas

pressure control in each of the MWPC's and performs calibration operations for both spectrometers. From a software standpoint, these functions are identical to those described for the Large Area Proportional Counter Array. In total, the instrument control subroutine would use four standard CAMAC software modules.

The data acquisition subroutine would be written to correspond to the specific hardware configuration of CAMAC modules used for the spectrometer. Its operational concepts and functions provided, however, would be identical to those discussed for the Large Area Proportional Counter Array. A total of four standard CAMAC software modules would be used for the data acquisition subroutine.

The observation program would also be similar in functional concept to that for the previous instrument. Any on-line data reduction and quick-look displays would, of course, be specifically tailored for this instrument. In addition, the observation program for this instrument would have to provide for the coordination of the crystal observation cycles and the Spacelab attitude control.

High-Resolution Gamma-Ray Ge(Li) Spectrometer - The software requirements for the High-Resolution Gamma Ray Ge(Li) Spectrometer are not as extensive as those for the preceding two instruments, although the basic concepts are the same. The only function performed by the instrument control subroutine is the maintenance of the proper temperature environment for the optimum performance of the solid state detector. This subroutine requires two standard CAMAC software modules.

The data acquisition subroutine again responds on an event-by-event basis as for the previous two instruments. It uses three standard CAMAC software modules. The observation program deals primarily with the data acquisition subroutine since the temperature control provided by the instrument control subroutine is a continuous, unchanging activity except for periodic assessment of housekeeping data.

3.5.2.3 CAMAC Module Software Requirements

The X-Ray/Gamma-Ray Pallet requirements for CAMAC module level subroutines are summarized in Table 3-2. A total of ten different module subroutines are used. As described in the SIRTf payload discussion, each

Table 3-2. X-Ray/Gamma-Ray Pallet Payload CAMAC
Software Module Requirements

CAMAC Module	Instrument		
	Large-Area Proportional Counter Array	Bragg Crystal Spectrometer	High-Resolution Gamma-Ray Spectrometer
Scaler	•	•	•
Position Encoder	•	•	
Input Register	•	•	•
Output Driver	•	•	
Stepping Motor Controller	•	•	
Time Digitizer	•	•	
ADC's			
DC level	•	•	•
low-resolution pulse	•	•	
high-resolution pulse			•
DAC			•

subroutine is used on a shared basis for all hardware modules of that same type. Thus, although over 100 CAMAC hardware modules are utilized for this payload, the ten standard subroutines listed in the table, combined with the single standard CAMAC driver, satisfy all software interface requirements.

3.5.2.4 CAMAC Software Applicability Summary

As was found in the case of the SIRTf payload, a significant programming burden, and consequently cost, can be removed from the individual investigator using the X-Ray/Gamma-Ray Pallet if NASA supplies standard CAMAC software modules corresponding to the hardware utilized. In this way, the individual investigator is freed from the hardware/software interface manipulation requirements and can concentrate on his observation mode program and its associated data acquisition and instrument control subroutines.

APPENDIX I
CAMAC SOFTWARE PRODUCTS GUIDE

(Reproduced from Issue No. 14, CAMAC Bulletin, December 1975)

CAMAC SOFTWARE PRODUCTS GUIDE

INTRODUCTION

The Software Products Section of the CAMAC Products Guide lists a number of software packages, programs and routines which have been developed by software firms, manufacturers of CAMAC equipment, and at research laboratories.

Work is going on to implement IML — the intermediate level CAMAC language. One contribution to IML implementation is listed below, but at least five other laboratories are at present engaged in implementing IML on several computers.

The products listed below are either in current use or will be so in the nearest few months. Some

of the software listed is commercially available, information about other is presumably available from respective authors. The correctness of each entry has been carefully checked against data provided.

Inclusion in the list does not necessarily indicate endorsement, recommendation or approval by the ESONE Committee, nor does omission indicate disapproval.

The classification used tentatively and reproduced below, is the same as was proposed in the March 1974 issue (No. 9) of this Bulletin.

SOFTWARE CLASSIFICATION GROUPS

	Page		Page
.5	Software.	.54	Support Software I (translators).
.50	Fundamental Concepts, General Subjects.	.541	Assemblers (with/without macros).
.500	General Descriptions, Documentation, etc.	.542	Cross-Assemblers, Cross-compilers.
.501	Languages.	.543	Compilers.
.502		.544	Interpreters, Algorithms.
.51	User-Oriented Programs I (full system support with user run-time and CAMAC system service programs).	.55	Support Software II.
.52	User-Oriented Programs II (specific run-time programs).	.551	Loaders.
.53	User-Oriented Programs III (subprograms, routines, Hardware programs).	.552	Linking Programs.
		.553	Utility Routines.
		.57	Other Service Programs.
		.571	Editors.
		.572	Debugging Routines.
		.573	Test Routines.

.50 Fundamental Concepts, General Subjects

<p>READER SERVICE CLASS CODE = TITLE = . . . AUTHOR(S) = PUBL, REF, =</p>	<p>REF NO 14,5001 .50 IMPLEMENTING CAMAC BY COMPILERS W. KNEIS, GFK, ZYKLUTRON-LR,, KARLSRUHE, GERMANY PROC CAMAC SYMPOS, LUXMBG, DEC 1973</p>	<p>DESCRIPTION = DEMANDS ON REAL-TIME SYSTEMS SUCH AS MINIMUM EXECUTION TIME MINIMUM CODE REQUIREMENTS, ETC., RECOMMEND THE USE OF COM- PILERS IN PROGRAMMING. THE POSSIBILITY TO IMPLEMENT A CAMAC LANGUAGE BY A COMPILER IS FIRST OF ALL A FUNCTION OF THE LEVEL AND CONCEPT OF THE LANGUAGE. META-LANGUAGES, THE SYN- TAX OF A PROGRAMMING LANGUAGE, ARE USED TO FORMULATE A COM- PILER FOR A SPECIFIC LANGUAGE. THE METHOD DESCRIBED HAS BEEN USED TO WRITE A COMPILER FOR IML, THE INTERMEDIATE LEVEL CAMAC LANGUAGE, IMPLEMENTED IN AN ASSEMBLER ENVIRONMENT.</p>
<p>READER SERVICE CLASS CODE = TITLE = . . . AUTHOR(S) = PUBL, REF, = ESONE REGISTR DATE</p>	<p>REF NO 14,5002 .50 PROCEDURE CALLS = A PRAGMATIC APPROACH J. MICHELSON, M. HALLING, KFA, JUELICH PROC CAMAC SYMPOS, LUXMBG, DEC 1973 31 MAY 1974</p>	<p>DESCRIPTION = DISCUSSION OF PROCEDURE CALLS AS THE BASIS FOR CAMAC SOFTWARE WITHIN HIGH-LEVEL LANGUAGES. COMPARISON WITH SYNTAX MODIFI- CATIONS TO LANGUAGES. DISCUSSION OF IMPLEMENTATION RESTRICTIONS DUE TO LANGUAGE REQUIREMENTS FOR EXISTING HIGH- LEVEL LANGUAGES, E.G. CLOSED SYSTEM-SUBROUTINES WHICH EXE- CUTE ONE DEFINED OPERATION (INVOLVING ONE OR MORE CAMAC CYCLES AS A GROUP). COMPARISON OF US-NIM CAMAC FURTHER SUBROUTINES AND PROCEDURE-CALL SYNTAX OF ESONE S/W IML LANGUAGE. APPLICATION OF PROCEDURE-CALLS TO APPLICATION- ORIENTED SOFTWARE.</p>
<p>READER SERVICE CLASS CODE = TITLE = . . . AUTHOR(S) = PUBL, REF, = NAME/ACRONYM = OPERATIVE DATE- COMPUTER = INTERFACE(S) = SOFTWARE TYPE = INCOMP TECHNIQUE FACILITIES =</p>	<p>REF NO 14,5003 .501(PL=11) CAMAC FACILITIES IN THE PROGRAMMING LANGUAGE OF PL=11 ROBERT D RUSSELL, CERN, GENEVA PROC CAMAC SYMPOS, LUXMBG, DEC 1973 YELLOW REPORT, CERN 74-24, DEC 1974 EXTENDED PL=11 1971/72 POP=11, WORD LENGTH 16 BITS CA=11 (EGEGURTEC) LANGUAGE, PL=11(EXTENDED) IN-LINE CODING OF CAMAC STATEMENTS SYMBOLIC DEVICE NAME USED DEMAND HANDLING IS INCLUDED</p>	<p>DESCRIPTION = PL=11 IS AN INTERMEDIATE-LEVEL, MACHINE-ORIENTED PROGRAMMING LANGUAGE EXTENDED TO INCLUDE CAMAC FEATURES. SYNTACTIC FORM OF CAMAC STATEMENTS ARE ANALOGOUS TO STANDARD PL=11 STATE- MENTS. SYMBOLIC NAMES FOR VARIABLES AND FUNCTIONS ARE DE- CLARED AT ONCE, AND OPERATIONS ARE EXECUTED BY STATEMENTS REFERRING TO THESE NAMES. USE OF SYMBOLIC NAMES MAKES PRO- GRAMS READABLE, AND SIMPLIFIES MODIFICATIONS OF CAMAC CON- FIGURATIONS. EXAMPLE OF STANDARD STATEMENT = WHILE PRINTSTATUS = BUSY DO EXAMPLE OF CAMAC STATEMENT = WHILE CRTSTATUS = BUSY DO</p>
<p>READER SERVICE CLASS CODE = AUTHOR(S) = NAME/ACRONYM = COMPUTER = SOFTWARE TYPE =</p>	<p>REF NO 14,5004 .501 (CATY) F R GOLDING, DARTSBURY LABORATORIES CATY ANY LANGUAGE (BASED ON BASIC)</p>	<p>DESCRIPTION = CATY IS A MACHINE INDEPENDENT HIGH-LEVEL LANGUAGE BASED UPON A SUBSET OF BASIC WITH EXTENSIONS FOR ADDRESSING CAMAC. PROGRAMS WRITTEN IN CATY ARE COMPILED AND NOT INTERPRETED. THUS, THE SPEED OF OPERATION WHEN CAMAC IS TESTED UNDER CATY IS COMPARABLE WITH THE SPEED OF OPERATION IN APPLICATIONS. CATY HAS BEEN IMPLEMENTED ON SEVERAL COMPUTERS (SEE .543).</p>
<p>READER SERVICE CLASS CODE = TITLE = . . . AUTHOR(S) = NAME/ACRONYM = OBTAINABLE FROM = AVAILABLE ON/AS = SOFTWARE TYPE =</p>	<p>REF NO 14,5005 .501 (CATY) SPECIFICATION OF THE LANGUAGE CATY C1030 R F CRANFIELD, GEC ELLIOTT (SEE ALSO PREVIOUS ENTRY) CATY GEC ELLIOTT (SEE LIST OF MANUFACTURERS) DESCRIPTION LANGUAGE (BASED ON BASIC)</p>	<p>DESCRIPTION = THE MAIN SPECIFICATION DESCRIBES THE FACILITIES AVAILABLE IN THE MACHINE INDEPENDENT HIGH LEVEL LANGUAGE CATY. APPENDICES TO THE SPECIFICATION DESCRIBE THE ADDITIONAL FEATURES ASSOCI- ATED WITH IMPLEMENTATIONS, ALL USING GEC ELLIOTT SYSTEM CRATE INTERFACES ON THE POP=11, NOVA, GEC-4080, AND GEC-2050 COMPUTERS.</p>
<p>READER SERVICE CLASS CODE = TITLE = . . . PREPARED BY = PUBL, REF, = NAME/ACRONYM = MAINTENANCE BY = OBTAINABLE FROM ESONE REGISTR DATE COMPUTER = SOFTWARE TYPE =</p>	<p>REF NO 14,5006 .501 (IML) THE DEFINITION OF IML A LANGUAGE FOR USE IN CAMAC SYSTEMS ESONE COMMITTEE, SOFTWARE W.G. AND AEC NIM COMMITTEE, SOFTWARE W.G. REPORT ESONE/IML/01, OCT 1974, AND REPORT IID=26615, JAN 1975 IML ESONE COMMITTEE IN COLLABORATION WITH NIM COMMITTEE ESONE SECRETARIAT AND U.S. GOVERN- MENT PRINTING OFFICE RESPECTIVELY AUG/SEPT 1974 ANY LANGUAGE</p>	<p>DESCRIPTION = IML IS A LANGUAGE USED TO EXPRESS THE OPERATIONS DESCRIBED IN THE CAMAC HARDWARE SPECIFICATIONS, AND THEIR INTERACTION WITH A COMPUTER SYSTEM. IML STATEMENTS LINK CAMAC STRUCTURES AND MODES OF OPERATION TO DATA STRUCTURES AND REAL-TIME FEATURES IN THE COMPUTER SYSTEM. THIS DEFINITION IS A GUIDE FOR THOSE IMPLEMENTING LANGUAGES AND OPERATING SYSTEMS WHO WISH TO MAKE CAMAC INPUT/OUTPUT AVAILABLE TO USERS. FEATURES ARE INCLUDED WHICH SUPPORT THE CAMAC BRANCH HIGHWAY AND THE CAMAC SERIAL HIGHWAY. THE LANGUAGE IS DEFINED SEMANTICALLY. THE SYNTAX USED TO EXPRESS IML DEPENDS ON THE ENVIRONMENT. THE MACRO SYNTAX IML=M1 IS DEFINED IN AN APPENDIX.</p>
<p>READER SERVICE CLASS CODE = TITLE = . . . AUTHOR(S) = NAME/ACRONYM = OBTAINABLE FROM = AVAILABLE ON/AS = SOFTWARE TYPE =</p>	<p>REF NO 14,5007 .501 (CASIC) A CAMAC EXTENDED BASIC LANGUAGE J M SERVET (SCHLUMBERGER) CASIC SCHLUMBERGER (SEE LIST OF MANUFACTURERS) DESCRIPTION LANGUAGE (EXTENDED BASIC)</p>	<p>DESCRIPTION = CASIC IS BASED ON BASIC AND PROVIDES ALL STANDARD STATEMENTS OF BASIC PLUS A SET OF CAMAC RELATED STATEMENTS. CASIC = LIKE BASIC = IS CONVERSATIONAL. THE MOST RECENT VERSION CONFORMS TO THE IML LANGUAGE (SEE .501(IML)) DEFINED BY THE ESONE COMMITTEE. CASIC IS IMPLEMENTED ON POP=11 (SEE .544).</p>

ORIGINAL PAGE IS
OF POOR QUALITY

.51 User-Oriented Programs I (full system support)

<p>READER SERVICE CLASS CODE = TITLE = - - -</p> <p>AUTHOR(S) = PUBL. REF. = NAME/ACRONYM = AVAILABLE UN/AS OPERATIVE DATE = COMPUTER = INTERFACE(S) = SOFTWARE TYPE = LANGUAGE = CAMAC FACILITIES</p>	<p>REF NO 14,5008 .51 CAMAC OPERATING SYSTEM FOR CONTROL APPLICATIONS DR B. MERTENS, IKP, KFA, JUELICH CAMAC BULLETIN NO 9, MARCH 1974 COS PAPER TAPE, ASCII CODE 1972 PDP-15, CORE REQUIREMENTS = 16K TYPE 2200 (BOREH) SYSTEM PROGRAM FORTRAN & MACRO-ASSEMBLER SYMBOLIC DEVICE NAMES USED, SINGLE & MULTIPLE ACTION PER INSTRUCTION, REAL/TIME DEMAND HANDLING INCORPORATED</p>	<p>DESCRIPTION = THE SYSTEM SOFTWARE PACKAGE PERMITS READ AND WRITE UP TO 100 MODULES, REAL-TIME TASKS MAY BE DEFINED ON-LINE, ABOUT 60 ELEMENTARY COMMANDS ARE PRE-DEFINED, SUCH AS-- =NAME MODULE/CR1, N=2, A=3/DEFINE SYMBOLIC NAME =READ MODULE/F=0 =WRITE MODULE J21/F=16 =DISAB MODULE/F=24 =DEFINE TASK/OPEN A TASK=DEFINITION =END/CLOSE TASK=HILL =AFTER 15 SECS TASK/EXECUTE USER=DEFINED TASK =15 SECS FROM NOW =SULL MODULE 3456/VALUE TO BE WRITTEN NEXT TO MODULE</p>
<p>READER SERVICE CLASS CODE = TITLE = - - -</p> <p>AUTHOR(S) = NAME/ACRONYM = AVAILABLE UN/AS OPERATIVE DATE = COMPUTER = INTERFACE(S) = MIN SYSTEM CONFIG SOFTWARE TYPE = LANGUAGE =</p>	<p>REF NO 14,5009 .51 BACKGROUND-FORGROUND SYSTEM FOR PULSE-HEIGHT ANALYSIS OF TWO- DIMENSIONAL MULTILINE PROPORTIONAL CHAMBER DATA DR A. MEUSLER, IKP, KFA, JUELICH BFG PAPER TAPE, ASCII CODE 19747 PDP-15, CORE REQUIREMENTS = 24K TYPE 2200 (BOREH) MAGTAPE, DECTAPE, DISK, & MEMORY SCANNING DISPLAY (IN-HOUSE) SYSTEM PROGRAM FORTRAN & MACRO-ASSEMBLER</p>	<p>DESCRIPTION = THE SYSTEM SOFTWARE PERMITS START AND STOP OF BLOCK TRANSFER FROM THE A/D CONVERTERS TO THE PDP-15 MEMORY (LIST MODE OUTPUT ONTO MAGTAPE ON-LINE SORTING IF DESIRED). THE BOREH INTERFACE HAS BEEN MODIFIED TO ALLOW BLOCK LENGTHS UP TO 4K 18 BIT WORDS.</p>
<p>READER SERVICE CLASS CODE = TITLE = - - - AUTHOR(S) =</p> <p>PUBL. REF. = OPERATIVE DATE = COMPUTER = INTERFACE(S) = SOFTWARE TYPE =</p>	<p>REF NO 14,5010 .51 TRIUMF CONTROL SYSTEM SOFTWARE C. P. GURD, W. K. DAWSON, TRIUMF, UNIVERSITY OF ALBERTA, CANADA CAMAC BULLETIN NO 8, NOVEMBER 1972 1973 4 SUPERNOVAS IN-HOUSE TYPE FULL SYSTEM SUPPORT FOR CONTROL OF TRIUMF CYCLOTRON</p>	<p>DESCRIPTION = THE SYSTEM SOFTWARE PACKAGE MONITORS OVER 1000 ANALOGUE PARAMETERS AND 1000 DIGITAL STATUS POINTS, SEARCHES OUT-OF-LIMIT READINGS, DISPLAYS MEASUREMENTS ON REQUEST, SETS OVER 300 ANALOGUE POINTS FROM A CENTRAL CONSOLE AND PERFORMS A NUMBER OF OTHER ROUTINES. A REAL-TIME EXECUTIVE PROGRAM = NATS (FROM NUVA ASYNCHRONOUS TASKING SUPERVISOR) = SCHEDULES AND SUPERVISES CAMAC TASKS, SUPPORTED BY A SUBPROGRAM LIBRARY, AS THEY ARE REQUESTED, JOBS TO BE PERFORMED ARE STRUCTURED INTO SEQUENCES OF CAMAC OPERATIONS SPECIFIC TO A PIECE OF HARDWARE (= CAMAC MODULE). THERE IS THUS A DIRECT MODULAR HARDWARE=SOFTWARE CORRESPONDENCE. CONTROL IS BASICALLY CLOCK-INITIATED SOFTWARE SCAN OF CYCLOTRON MONITORING, BUT INTERRUPTS ARE INCLUDED, MAINLY INITIATED BY CONSOLE.</p>
<p>READER SERVICE CLASS CODE = TITLE = - - - NAME/ACRONYM = OBTAINABLE FROM = SOFTWARE TYPE = COMPUTER = INTERFACE(S) = HARDWARE CONFIG</p>	<p>REF NO 14,5011 .51 BASIC SINGLE PARAMETER MCA SYSTEM (MISP) MISP NUCLEAR ENTERPRISES (SEE INDEX OF MFERS) SYSTEM SOFTWARE PDP-11, 8K MEMORY & REAL TIME CLOCK 9030 (NUCL. ENTERPR) (PROGRAMMED TRANSFERS & INTERRUPT ONLY) ADC (LAREN OR 9060), 9021 LIVE TIME RTC, TTY/READER (7064), TEK603/604 OR LANSCOPE</p>	<p>DESCRIPTION = THE PROGRAM OCCUPIES 2K OF MEMORY AND USES A DATA AREA OF 4K FOR UP TO 4096 CHANNELS ACQUISITION. THE PACKAGE CONSISTS OF A DISPLAY DRIVER, A USER ORIENTED TELETYPE MANIPULATOR, ACQUISITION CONTROL, AND A DATA MANIPULATION ROUTINE. THE DISPLAY DRIVER IS RUN AS A BACKGROUND TASK WHICH IS INTERRUPTED BY THE ADC, CLOCKS AND TELETYPE. THIS PACKAGE CAN BE OBTAINED WITH MULTISCALE OPTION, THE HARDWARE IS EXTENDED WITH A 900J UN 003 SCALER, DATA AREA IS DIVIDED INTO 4 AREAS, EACH ONE THOUSAND CHANNELS.</p>
<p>READER SERVICE CLASS CODE = TITLE = - - - NAME/ACRONYM = OBTAINABLE FROM = SOFTWARE TYPE = COMPUTER = INTERFACE(S) = HARDWARE CONFIG</p>	<p>REF NO 14,5012 .51 DUAL MCA SYSTEM (DAMCAS) DAMCAS NUCLEAR ENTERPRISES (SEE INDEX OF MFERS) SYSTEM SOFTWARE PDP-11, 16K MEMORY & REAL TIME CLOCK 9030 & 9033 (NUCL. ENTERPR) (PROGRAMMED & AUTONOMOUS TRANSFERS) ADC (LAREN OR 9060), 9021 LIVE TIME RTC, TTY/READER (7064), MUNCH (7065), MAGTAPE (CS 0042), TEK603/604 OR LANSCOPE</p>	<p>DESCRIPTION = THE PROGRAM OCCUPIES 6K LEAVING 10K OF MEMORY FOR DATA ACQUISITION (4K OF 16 BITS & 4K OF 24 BITS). THE SOFTWARE PACKAGE CONSISTS OF A DISPLAY DRIVER, A TELETYPE HANDLER FOR OPERATION CONTROL OF DATA ACQUISITION, DATA MANIPULATION ROUTINE, AND A ROUTINE FOR AUTONOMOUS CONTROL OF DATA ACQUISITION AND MAG TAPE TRANSFERS.</p>
<p>READER SERVICE CLASS CODE = TITLE = - - - NAME/ACRONYM = OBTAINABLE FROM = SOFTWARE TYPE = COMPUTER = INTERFACE(S) = HARDWARE CONFIG</p>	<p>REF NO 14,5013 .51 MULTI PARAMETER DATA ACQUISITION SYSTEM MUDAS I NUCLEAR ENTERPRISES (SEE INDEX OF MFERS) SYSTEM SOFTWARE PDP-11, 8K MEMORY & REAL TIME CLOCK 9030 (NUCL. ENTERPR) (PROGRAMMED TRANSFERS & INTERRUPT ONLY) ADC'S (LAREN OR 9060) & CLINE SELECTOR (CS 0045), 9021 LIVE TIME RTC, TTY & MAG TAPE, TEK 603/604.</p>	<p>DESCRIPTION = THE SYSTEM IS CAPABLE OF ACCEPTING FIVE PARAMETER EVENTS AND STORING THEM ON MAG TAPE, SIMULTANEOUSLY PERFORMING MULTI-CHANNEL ANALYSIS ON ONE SELECTED PARAMETER. WINDOWS MAY BE SET ON EACH PARAMETER FOR BOTH MODES, TOGETHER WITH A COUNT DIVISION FACTOR SET OVER THE REGION OF INTEREST, DATA DUMPED IN LIST MODE MAY BE READ BACK FOR ANALYSIS.</p>

READER SERVICE
CLASS CODE =
AUTHOR(S) =
NAME/ACRONYM =
OPERATIVE DATE =
SOFTWARE TYPE =

REF NO 14,5014
.51
D GURD, TRIUMF, UNIV, ALBERTA, CANADA
CAMAC
DEC
1973
SYSTEM SOFTWARE

DESCRIPTION =
THE SYSTEM SOFTWARE = CAMAC = CONSISTS OF SEVERAL SUBROUTINE
CALLS, THESE ARE =
PRIMITIVE SUBROUTINES PERFORMING THE ACTUAL I/O OPERATIONS,
MODULE SUBROUTINES, THE MUX/ADC SUBROUTINES, CAMAC LAMB ON
INTERUPTS, SERIAL TASKS, AND AN INTERRUPTER (FOR DATA).

.52 User-Oriented Programs II (specific run-time programs)

READER SERVICE
CLASS CODE =
TITLE =
NAME/ACRONYM =
MAINTENANCE BY =
OBTAINABLE FROM
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
SOFTWARE TYPE =

REF NO 14,5015
.52
OPERATING SYSTEM SOFTWARE PACKAGES
SEE DESCRIPTION
DEC
DEC (SEE INDEX OF MANUFACTURERS)
1975
PDP-11
SEE DESCRIPTION
CAMAC SERVICE ROUTINES, USER,
INTERFAC = & DESCRIPTOR PROGRAMS

DESCRIPTION =
THE SOFTWARE PACKAGES ARE COMPLETE OPERATING SYSTEMS,
CONTROLLERS AND OPERATING SYSTEMS ARE RELATED AS FOLLOWS =
CA-11=C USES RDX=11=NO OPERATING SYSTEM
CA-11=E USES RDX=11=M UM RI=11
CA-11=P USES RDX=11=M UM RI=11

READER SERVICE
CLASS CODE =
TITLE =
NAME/ACRONYM =
OBTAINABLE FROM
SOFTWARE TYPE =
COMPUTER =
INTERFACE(S) =

REF NO 14,5016
.52
CASMAC = A SOFTWARE PACKAGE FOR COMMUNI-
CATION WITH CAMAC=PROCESS=PERIPHERALS
CASMAC
IDAS (SEE INDEX OF MANUFACTURERS)
SYSTEM OF RE-ENTRANT ASSEMBLER ROUTINES
PDP-11 (DEC), MIN 740 WORDS OF MEMORY
1 CP-11 (SCHLUMBERGER)

DESCRIPTION =
THE SYSTEM OF ASSEMBLER ROUTINES ALLOW COMMUNICATION WITH
CAMAC=PROCESS=PERIPHERALS USING SINGLE=WORD TRANSFER MODE AS
WELL AS BLOCK TRANSFER MODE ON FURTHER AND ASSEMBLER LEVEL,
INTERUPT ACTIONS CAN BE OBTAINED IN THE FORM OF AN ARBITRARY
SEQUENCE OF CAMAC TRANSFERS ON FURTHER LEVEL,
NO SOFTWARE OPERATING SYSTEM IS NEEDED, AND CASMAC CAN
THEREFORE BE USED AUTONOMOUSLY AS WELL AS IN CONNECTION WITH
A REAL TIME OR BATCH OPERATING SYSTEM.

.53 User-Oriented Programs III (subprograms, etc.)

READER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
PUBL, REF, =
NAME/ACRONYM =
MAINTENANCE BY =
OBTAINABLE FROM
AVAILABLE UN/AS
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
MIN SYSTEM CONFIG
SOFTWARE TYPE =
LANGUAGE =
HOST LANGUAGE =
CAMAC FACILITIES
FACILITIES =

REF NO 14,5017
.53 (BASIC)
CAMAC AND INTERACTING PROGRAMMING
DR E H HIMMER, CEHN, GENEVA
PROC CAMAC SYMPOS, LUXMBG, DEC 1973
& BASIC CALLABLE ROUTINES,
NP GROUP NOTE, NP=DMG, CEHN
HPCRA, HPCMB, HPCMC
DR E H HIMMER
NP DIV, CEHN, CH=1211 GENEVA
PAPER TAPE, ASCII CODE
1971/72
M=P 2100=SERIES, 8K 16 BIT WORDS
2201(BOMER), 7218 & HPC=060(CEHN)
TTY ON TEK 4010 TERMINAL & CC=1
SET OF SUBROUTINES
HP ASSEMBLY
BASIC (NP EXTENSION OF)
IN-LINE CODED CALLS IN BASIC,
SUBROUTINES IN ASSEMBLY, HAS ADDN
SINGLE & MULTIPLE ACTION PER
INSTRUCTION, NO DEMAND HANDLING

DESCRIPTION =
THESE BASIC=CALLABLE CAMAC SUBROUTINES IN THREE VERSIONS FOR
THREE INTERFACES PROVIDE MUST COMMAND FACILITIES FOR CONTROL
AND DATA TRANSFER, DATA WORDS MAY BE 16 OR 24 BITS LONG
(ONLY 16 BITS FOR HPC=060), BINARY, BCD OR LOGIC (0 OR 1),
ROUTINES COVER BLOCK TRANSFERS, PROGRAMMED AND SEQUENTIAL
ADDRESSING & UTILITY ROUTINES, IN TOTAL 18 & 3 OPTIONALLY,
GENERAL FORM OF CALL STATEMENT =
= CALL (SUBROUTINE NUMBER, C, N, A, F, U, W)
= CALL (SUBROUTINE NUMBER, C, N, A, F, U, W, W)
WHERE W IS WORD COUNT, D IS DATA, C, N, A, F, & W HAVE USUAL
MEANING
EX= CALL(10,1,2,0,16,U(1),U,20)
TIME IS APPROX 5 MSEC/STATEMENT, BLOCK TRANSFER CALL GENE-
RATED DIRECTLY BY INTERFACE ARE MUCH FASTER.

READER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
PUBL, REF, =
NAME/ACRONYM =
OBTAINABLE FROM
AVAILABLE UN/AS
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
SOFTWARE TYPE =
LANGUAGE =
CAMAC FACILITIES

REF NO 14,5018
.53(FORTRAN)
SPECIFICATIONS FOR STANDARD CAMAC
SUBROUTINES
RICHARD F THOMAS JR,
CAMAC BULLETIN NO 6, MARCH 1973
SEE DESCRIPTION
USAC NIM COMMITTEE, CAMAC SWG
ALGORITHM
1973
INDEPENDENT, MEMORY SIZE NOT SPEC,
ANY
SET OF SUBROUTINES
FORTRAN
FUNDAMENTAL CAMAC OPERATIONS, STANDARD
BLOCK TRANSFERS IN SINGLE & MULTIPLE
ACTION STATEMENTS

DESCRIPTION =
A SET OF 6 SUBROUTINES, OF WHICH ONE IS CALLED BY ALL THE
OTHER PERMITS A GREAT VARIETY OF SINGLE AND MULTIPLE CAMAC
OPERATIONS TO BE PERFORMED, DEMAND HANDLING, OTHER THAN BY
TEST LAM, IS NOT COVERED,
THE SUBROUTINES EXECUTE CAMAC OPERATIONS AS FOLLOWS =
CMCASC = SINGLE CAMAC FUNCTION AT SINGLE ADDRESS
ONE OR MORE TIMES
CMCSEQ = SINGLE CAMAC FUNCTION AT SUCCESSION OF ADDRESSES
CMCASC = SPECIFIED CAMAC FUNCTION IN ADDRESS SCAN MODE
CMCMT = SPECIFIED CAMAC FUNCTION IN REPEAT MODE
CMCSTP = SPECIFIED CAMAC FUNCTION IN STOP MODE
CMCLUP = SPECIFIED CAMAC FUNCTION AT A HIERARCHICAL SEQUENCE
OF ADDRESSES WITH OPTIONAL SKIP OF SEQUENCE BASED ON Q,
GENERAL FORM OF STATEMENT =
CALL CMC,, (PARAMETER LIST)
EXAMPLE = CALL CMCSTP (F,B,C,N,AU,LN,DATA,ERRORA,NEX)

READER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
NAME/ACRONYM =
VERSION =
OBTAINABLE FROM
AVAILABLE UN/AS
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
SOFTWARE TYPE =
LANGUAGE =
INCURP TECHNIQUE
CAMAC FACILITIES

REF NO 14,5019
.53(FORTRAN)
FORTRAN SUBROUTINES
M PUEL
FORTRAN CALLS
V002
M PUEL, ZEL, KFA, JUEFICH
DECTAPE
MARCH 1972
PDP-11, 16K 16 BIT WORDS MEMORY
TYPE 1533A (BOMER)
PROCEDURE CALLS
FORTRAN (ON PDP-11 (THREADED CODE)
IN-LINE SUBROUTINE CALLS
SINGLE ACTION STATEMENTS

DESCRIPTION =
FORTRAN SUBROUTINES FOR SINGLE ACTIONS, MUCH SIMPLER THAN
THE NIM APPROACH (REF, M, F, THOMAS) FOR THE BOMER 1533A
CONTROLLER WRITTEN IN RE-ENTRANT CODE.

HEADER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
NAME/ACRONYM =
VERSION =
MAINTENANCE BY =
OBTAINABLE FROM =
AVAILABLE UN/AS =
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
SOFTWARE TYPE =
LANGUAGE =
HOST LANGUAGE =
INCORP TECHNIQUE =
CAMAC FACILITIES =

REF NO 14,5020
.53
CAMAC FUNCTION FOR RT11
L BYARS, R KEYSER
CAMAC, CAMINT
RT11
ORTEC
ORTEC (SEE INDEX OF MANUFACTURERS)
PAPER TAPE
1974
PDP-11
DC011 (EG&G/ORTEC)
SUBROUTINES
PDP-11 ASSEMBLY
RT11/FORTRAN
CALLS TO FORTRAN LIBRARY ROUTINES
SINGLE OR MULTIPLE INSTRUCTIONS,
DEMAND HANDLING

DESCRIPTION =
THIS SOFTWARE PACKAGE CONSISTS OF A NUMBER OF SUBROUTINES
FOR FORTRAN/RT11 CALLING CAMAC FUNCTIONS.
THE CAMAC CALL STATEMENT HAS THE FOLLOWING FORM--
CALL CAMAC (IF, IN, IA, IO, IDATA)
THEY ARE USED TO TRANSFER DATA TO/FROM CAMAC AND FOR TEST
PURPOSES.
IF, IN, IA ARE RESPECTIVELY FUNCTION, STATION ADDRESS AND
SUBADDRESS, IO IS BOTH UNIT AND UNIT,
CAMINT IS USED TO HANDLE INTERRUPTS FROM CAMAC CHATE, AND
HAS THE GENERAL FORM--
CAMINT(IN,NAME1)
WHERE IN IS THE STATION NUMBER AND NAME1 IS THE NAME OF THE
SUBROUTINE TO BE EXECUTED WHEN THE INTERRUPT OCCURS.

HEADER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
NAME/ACRONYM =
OBTAINABLE FROM =
AVAILABLE UN/AS =
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
LANGUAGE =
SOFTWARE TYPE =

REF NO 14,5021
.53(FORTRAN)
J M STEPHENSON, L A KLAISNER
KSLCIR
KINETIC SYSTEMS (SEE INDEX OF MFMS)
1974
PDP-11, 16K CORE MEMORY REQUIRED
TYPES 3911A, 3991 & 3992 (KINETIC)
FORTRAN
LIBRARY OF FORTRAN FUNCTIONS AND
SUBROUTINES

DESCRIPTION =
THIS SOFTWARE PACKAGE IMPLEMENTS THE CMCHSC SERIES OF STAND-
PROGRAM AND FORTRAN CALLS DESCRIBED IN CAMAC BULLETIN NO 6, 1973.
IT ALSO INCLUDES THE BIT MANIPULATION FUNCTIONS EXCLUSIVE
ON, INCLUSIVE ON, AND, NOT, & SHIFT. THE PACKAGE SUPPORTS
UP TO 8 CHATES INTERFACED THROUGH MODEL 3911A UNIBUS *J
CHATE CONTROLLERS, UP TO 7 CHATES PER 3991 BRANCH DRIVER AND
UP TO 61 CHATES PER 3992 SERIAL BRANCH DRIVER. THE NUMBER
OF PARALLEL AND SERIAL BRANCHES SHOULD BE LESS THAN 8.
*) UNIBUS IS A TRADE MARK OF DIGITAL EQUIPMENT CORP.

HEADER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
NAME/ACRONYM =
OBTAINABLE FROM =
AVAILABLE UN/AS =
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
MIN MEMORY SPACE =
MIN SYSTEM CONFIG
SOFTWARE TYPE =
ENVIRONMENT FOR =
LANGUAGE =
FACILITIES =

REF NO 14,5022
.53
I/O MACROS FOR CAMAC
D STUCKENBRUCK, G KLEINERT,
SIEMENS AG, KARLSRUHE
MACAM
SIEMENS (SEE INDEX OF MFMS)
PAPER TAPE, CARDS & SOURCE DECK
NOVEMBER 1974
PR 320/330
CC 320 & SC 330 (SIEMENS)
.5K = 1K OF 16 BITS (SUPERVISOR EXCL)
DEPENDENT ON HARDWARE
TTY AND SUPERVISOR PROGRAM
I/O ROUTINES, LAM HANDLING
CAMAC SOFTWARE IS ASSEMBLER 300
MACROS = ASSEMBLER CALLS = FORTRAN
CONCURRENT MULTI-USER OPERATION, SYSTEM
RUNS UNDER REAL-TIME SUPERVISOR

DESCRIPTION =
A SET OF I/O MACRO SUBROUTINES CAN BE CALLED BY ANY USER
PROGRAM CONCURRENTLY RUNNING ON THE COMPUTER, PROVIDED THEY
OPERATE UNDER A REAL-TIME SUPERVISOR PROGRAM. THE ROUTINES
COMPRISE THE FUNCTIONS READ, WRITE, AND EXECUTION OF CONTROL
COMMANDS. BLOCK TRANSFERS ARE PERFORMED ON CONSTANT OR
VARIABLE CAMAC ADDRESS, AND IN INCREMENT MODE OR HANDOFF-LIST
MODE. THE COORDINATION OF USER PROGRAMS AND CAMAC PROVIDED
BY THE SUPERVISOR, FACILITATES GREATLY THE LAM HANDLING.
THE SYSTEM ALLWS UP TO 8 BRANCHES, EACH WITH 7 CHATES.
SYSTEM SOFTWARE ENVIRONMENTS FACILITATE INCORPORATION OF
THE SUBROUTINE CALLS AS STATEMENTS EMBEDDED IN FORTRAN
PROGRAMS.

HEADER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
NAME/ACRONYM =
OBTAINABLE FROM =
AVAILABLE UN/AS =
OPERATIVE DATE =
COMPUTER =
MIN MEMORY SPACE =
INTERFACE(S) =
SOFTWARE TYPE =
MIN SYSTEM CONFIG
INCORP TECHNIQUE =
CAMAC FACILITIES =

REF NO 14,5023
.53 (BASIC)
BASIC = SUBROUTINES
D STUCKENBRUCK, SIEMENS AG, KARLSRUHE
BASIC = CALLS
SIEMENS (SEE INDEX OF MANUFACTURERS)
PAPER TAPE, CARDS
1973
PR 320
1K OF 16 BITS (BASIC COMPILE EXCLUDED)
CC 320
SUBROUTINES
TTY AND BASIC COMPILE
EMBEDDED BASIC CALLS TO SUBROUTINES
LAM HANDLING

DESCRIPTION =
THE SUBROUTINES IN ASSEMBLER ARE HANDLED BY THE BASIC-UN-320
COMPILE (INTERPRETER?).
THE STATEMENT =
CALL (CM, PARAMETER LIST)
CAUSES PROGRAM TO JUMP TO SUBROUTINE CALLED.
THE FOLLOWING CAMAC OPERATIONS CAN BE EXECUTED =
= SINGLE OPERATION (READ, WRITE, CONTROL)
= INTERRUPT REGISTRATION AND JUMP TO LAM HANDLING ROUTINE
= WAITING FOR LAM
PARAMETER LIST IS A STRING OF CHARACTERS SPECIFYING THE
OPERATION TO BE EXECUTED.
EXAMPLE =
CALL(CM,NAP,11,0,0,X1)
= WHERE 11,0,0, = STATION,SUBADDRESS,FUNCTION, X1 = VARIABLE

HEADER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
NAME/ACRONYM =
OBTAINABLE FROM =
AVAILABLE UN/AS =
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
OPERATING SYSTEM
SOFTWARE TYPE =
LANGUAGE =
HOST LANGUAGE =
INCORP TECHNIQUE =
CAMAC FACILITIES =

REF NO 14,5024
.53
TWO-LEVEL CAMAC PERIPHERAL HANDLER
L M TAFF, UNIV OF GROMINGEN, NETHERLANDS
COMPUTER PHYSICS COMMUNICATIONS
(TO BE PUBLISHED)
AUTHOR
AUTHOR
DECTAPE (ASCII CODE)
1974
DEC PDP-11, MIN 8K OF MEMORY
CA-15 (DEC)
(SOFTWARE) = DEC MONITOR SYSTEM (ADSS)
CAMAC DRIVER/LAM HANDLER SUBROUTINE,
I/O DEVICE HANDLERS, CMCHSC SUBROUTINE
ASSEMBLER
ANY SUPPORTED BY SYSTEM
LINKED AT LOAD TIME
SINGLE CAMAC OPERATIONS, DATA CHANNEL
TRANSFERS, DEMAND HANDLING, RE-ENTRANT

DESCRIPTION =
THE CAMAC DRIVER/LAM HANDLER IS A GLOBALLY LINKED SUBROUTINE
FOR EXECUTING SINGLE CAMAC OPERATIONS, CONTROLLING ACCESS TO 2
HARDWARE DATA CHANNELS VIA BUFFERS, AND GIVING CONTROL TO THE
PROPER USER ROUTINE WHEN A LAM OCCURS. IT MAY BE CALLED BY
ASSEMBLER LOADED USER PROGRAMS, THOMAS STANDARDS SUBROUTINE
CMCHSC (HENCE ALL OTHER OF HIS ROUTINES WHICH CALL CMCHSC)
= SEE .53 ABOVE = AND I/O HANDLERS FOR CAMAC INTERFACED
PERIPHERALS, EITHER FROM MAINSTREAM OR LAM HANDOFF PRIOITY.
CAMAC INTERFACED DEVICES FOR WHICH HANDLERS CURRENTLY EXIST
INCLUDE A LINE PRINTER, CARD READER, INCREMENTAL PLUITER, AND
A TEKTRONIX 4010 TERMINAL. FOR DEVICE HANDLERS, CAMAC IS
TRANSPARENT.
IT IS RELATIVELY EASY TO ADAPT A HANDLER FOR AN I/O BUS DEVICE
TO CAMAC SIMPLY BY SUBSTITUTING SUBROUTINE CALLS TO THE UNIVER-
SAL I/O OPERATIONS AND UNSERVING A FEW NONRESTRICTIVE CONVEN-
TIONS. THIS TWO-LEVEL APPROACH CAN ACCOMMODATE CAMAC LANGUAGES
IF ACTION STATEMENTS ARE COMPILED INTO SUBROUTINE CALLS.

HEADER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
NAME/ACRONYM =
OBTAINABLE FROM =
AVAILABLE UN/AS =
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
SOFTWARE TYPE =

REF NO 14,5025
.53
CAMAC/FORTRAN V INTERFACE SOFTWARE
A GSPUNNER, SEN ELECTRONIQUE
SEN
DISK (RDOS), FULL MDOS COMPATIBILITY
MAY 1975
ANY AGVA (WITH/OUT FLOATING POINT)
CC 2023 (SEN)
POST PROCESSOR

DESCRIPTION =

.54 Support Software I (translators)

HEADER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
MAINTENANCE BY =
OBTAINABLE FROM
OPERATIVE DATE =
SOFTWARE TYPE =
LANGUAGE =
COMPUTER =
CAMAC FACILITIES

REF NO 14,5026
.54
SYNOPSIS AN UNIVERSAL MACRO PROCESSOR
SOFTWARE PARTIALS
SOFTWARE PARTIALS
SAME, (SEE INDEX OF MANUFACTURERS)
APRIL 1974
MACRO PROCESSOR
WRITTEN IN HIGH LEVEL LANGUAGE
CAN RUN ON IBM, UNIVAC, CDC, ICL,
SIEMENS, ETC.
INCORPORATED IN-LINE FOR FULL-SET
IML WITH MACRO PROCESSOR DIRECTIVES

DESCRIPTION =
SYNOPSIS IS A LANGUAGE INDEPENDENT MACRO PROCESSOR AND
THEREFORE A TOOL FOR MACRO EXPANSION OF EXISTING OR
FUTURE PROGRAMMING LANGUAGE. THIS SYNOPSIS MAINTAINS AND
PROCESSES MACROS IN HIGH LEVEL LANGUAGES (FORTRAN, BASIC,
ALGOL, PL/I, ETC.) AS WELL AS ASSEMBLY LANGUAGES. SYNOPSIS
OPERATES AS A MACRO PROCESSOR GENERATING SOURCE CODE
STATEMENTS FOR SUBSEQUENT COMPILE, POSSIBLY ON ANOTHER
COMPUTER.

HEADER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
MAINTENANCE BY =
OBTAINABLE FROM
SOFTWARE TYPE =
INTERFACE(S) =

REF NO 14,5027
.541
A MACRO ASSEMBLER FOR TYPE MBD=11
MICROPROGRAMMED BRANCH DRIVE
PDP=11
BI RA SYSTEMS (SEE INDEX OF MHS)
MACRO ASSEMBLER (TRANSLATOR)
MBD=11 (BI RA SYSTEMS)

DESCRIPTION =
THE MACRO ASSEMBLER HAS BEEN DEVELOPED TO FACILITATE THE
WRITING OF PROGRAMS FOR THE MBD=11 MICROPROCESSOR INTERFACE.
THE ASSEMBLER TRANSLATES PROGRAMS WRITTEN IN MACRO CODE INTO
INSTRUCTIONS ACCEPTABLE BY THE MBD=11. UP TO 4K INSTRUCTIONS
CAN BE STORED IN THE MBD=11, A FUNCTION OF MEMORY SIZE
WHICH GO FROM 256 TO 4K WORDS IN INCREMENTS OF 256 AND 1K.
INSTRUCTIONS ARE MICROSTRUCTURED FORMING A POWERFUL SET.

HEADER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
NAME/ACRONYM =
MAINTENANCE BY =
OBTAINABLE FROM
AVAILABLE UN/AS
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
MIN SYSTEM CONFIG
SOFTWARE TYPE =
LANGUAGE =
CAMAC FEATURES =
ENVIRONMENT FOR =
CAMAC FACILITIES

REF NO 14,5028
.541(MACRO11)
MACROS FOR 1533A
MR. HEEM
MACRO 1533A
MR. HEEM
MR. HEEM, ZEL, KFA, JUELICH
DECTAPE
FEBRUARY 1973
PDP=11, MIN 8K 16 BIT WORDS
TYPE 1533A (BUREAU)
DDS V004, 008, 009
MACRO=SET
MACRO 11
ARE INCORPORATED IN-LINE
CAMAC SOFTWARE IS ASSEMBLER
SINGLE ACTION STATEMENTS,
SYMBOLIC DEVICE NAMES

DESCRIPTION =
THIS IS A SIMPLE MACRO SET (NO DECLARATIONS) FOR SINGLE
ACTION STATEMENTS. EXECUTION SPEED IS HIGH (APPROX 30
MICROSECS PER INSTRUCTION, DEPENDENT ON TYPE OF INSTRUCTION
IN TYPE OF PDP=11). NOT INTERRUPTABLE MACROS (PRIORITY=)

HEADER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
PUBL. REF. =
OBTAINABLE FROM
AVAILABLE UN/AS
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
MIN SYSTEM CONFIG
SOFTWARE TYPE =
LANGUAGE =
CAMAC FEATURES =
CAMAC FACILITIES

REF NO 14,5029
.541(IML)
MACRO=IML IMPLEMENTATIONS FOR DEC
PDP=11 COMPUTERS
M KUBITZ, R KINO, HMI=BERLIN
CAMAC BULLETIN NO 12, APRIL 1975
M KUBITZ, BEREICH D/E, HMI=BERLIN
GERMANY
ALL MEDIA
1974
PDP=11, 16K, 24K, 44K, OR 52K
CA=11A (DEC), 1533A (HUREN)
DDS V08/09, RSX=110, RSX=11M
MACRO SET OF IML (IMPLEMENTED)
PDP=11 ASSEMBLY
INCORPORATED BY MACROS
FULL SET OF IML=MACROS
INCLUDING DEMAND HANDLING

DESCRIPTION =
IML IS IMPLEMENTED ON PDP=11 IN ACCORDANCE WITH THE MACRO
SYNTAX AS DEFINED IN THE DOCUMENT E5096/IML/01 (SEE CLASS
501 ABOVE). VERSIONS ARE AVAILABLE FOR INTERFACE
CONTROLLERS AND DEC OPERATING SYSTEMS AS MENTIONED IN THE
LEFT COLUMN.
IMPLEMENTATION COVERS THE FULL SET OF IML MACROS AND DEMAND
HANDLING EXCEPT BLOCK TRANSFER ON SPECIAL LAP, X-EMPHASIS
CONTROL STATEMENTS, AND SUBSCRIPT MODE. TRANSFER MODES NOT
IMPLEMENTED BY HARDWARE ARE SIMULATED BY SOFTWARE.
I/O TRANSFER INSTRUCTIONS ARE EMBEDDED IN THE MACROS AND ARE
PERFORMED DIRECTLY IN ACTION BY THE MACROS.
ADDRESS CALCULATION AT ASSEMBLY TIME GIVES OPTIMIZED
ADDRESS CALCULATION AT ASSEMBLY TIME GIVES OPTIMUM RUN TIME
CODE. MOST LANGUAGES CAN BE PDP=11 MACRO ASSEMBLER OR
FORTRAN (VIA SUBROUTINE CALL).
MEMORY REQUIREMENTS VARY WITH OPERATING SYSTEM AND IF FULL
SET IS NEEDED, ON A SUB=SET IS ACCEPTABLE. 16K IS REQUIRED
FOR A SUB=SET WITH DDSV08/09 OR RSX=11M AND 52K FOR FULL SET
AND RSX=110.

HEADER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
NAME/ACRONYM =
OBTAINABLE FROM
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
MIN SYSTEM CONFIG
SOFTWARE TYPE =
LANGUAGE =

REF NO 14,5030
.543(CATY)
A CAMAC TESTING AID FOR USE ON PDP=11
F R GULDING, APPLIED COMPUTER SYST,
CAT11
APPLIED COMPUTER SYSTEMS LTD,
WENZEL ELEKTRONIK, NUCL ENTERPRISES,
(SEE INDEX OF MANUFACTURERS)
1973
PDP=11, 4K OR 8K MEMORY REQUIRED
DEPENDENT ON VERSION
C=CSC=11 (WENZEL), V030 (N.E.)
CONTROL VISTA, HEADEN, PUNCH
SYSTEM (EXECUTIVE, COMPILE ETC)
CATY (BASED ON BASIC)

DESCRIPTION =
USERS TEST PROGRAMS ARE TYPED IN AND THEREAFTER COMPILED AND
RUN. IT IS POSSIBLE TO EDIT THE PROGRAM AND RETURN IT WITH
OUT HAVING TO RETYPE THE ORIGINAL PROGRAM. CAMAC COMMANDS
ARE EMBEDDED IN PROGRAM AS STATEMENT LINES.
CAT11 HAS INTERRUPT AS SYSTEM FEATURE. THE USER MAY TYPE HIS
OWN INTERRUPT ROUTINE.
THE CAT11 EXECUTIVE PROGRAM CHANGES SLIGHTLY WITH INTERFACE
USED, BUT ALL ROUTINES ARE IDENTICAL.
VERSIONS OF THIS SYSTEM IS ALSO AVAILABLE FROM DEC ELLIOTT
(SEE FOLLOWING ENTRIES)

HEADER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
OBTAINABLE FROM
OPERATIVE DATE =
COMPUTER =
MIN MEMORY SPACE
INTERFACE(S) =
MIN SYSTEM CONFIG
LANGUAGE =

REF NO 14,5031
.543(CATY)
A CAMAC TESTING AID = CATY = FOR PDP=11
F R GULDING, R F CHANFIELD
DEC ELLIOTT (SEE INDEX OF MANUFACTURERS)
1974
PDP=11, MIN 4K REQUIRED
PTI=11C/D, IVG=11 (DEC ELLIOTT)
CONTROL TTY OR VISTA, HEADEN, PUNCH
CATY (BASED ON BASIC)

DESCRIPTION =
SEE PRECEDING ENTRY

ORIGINAL PAGE IS
OF POOR QUALITY

READER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
OBTAINABLE FROM =
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
MIN SYSTEM CONFIG
LANGUAGE =

REF NO 14,5032
.543(CATY)
A CAMAC TESTING AID = CATY = FUN MUVA
F R GULLING, W F CHANFIELD
GEC ELLIOTT
MARCH 1975
MUVA SERIES (DATA GENERAL), MIN 4K
MUVA EXECUTIVE SUITE (GEC ELLIOTT)
CONTROL TTY ON VDU, HEADEN, PUNCH
CATY (BASED ON BASIL)

DESCRIPTION =
(SEE CLASS .501(CATY) AND PRECEDING ENTRIES CLASS .543)

READER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
OBTAINABLE FROM =
COMPUTER =
INTERFACE(S) =
LANGUAGE =

REF NO 14,5033
.543(CATY)
A CAMAC TESTING AID = CATY = FUN THE
GEC ELLIOTT
2050 AND 4080 (GEC)
EXECUTIVE SUITE FUN 2050/4080 (GEC)
CATY (BASED ON BASIL)

DESCRIPTION =
(SEE CLASS .501(CATY) AND PRECEDING ENTRIES CLASS .543)

READER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
PUBL, REF, =
NAME/ACRONYM =
MAINTENANCE BY =
OBTAINABLE FROM =
AVAILABLE UN/AS
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
MIN SYSTEM CONFIG
SOFTWARE TYPE =
LANGUAGE =
INCRP TECHNIQUE
ENVIRONMENT FOR =
CAMAC FACILITIES

REF NO 14,5034
.543
A BASIC MACRO-11 COMPILER
B BECKS
CAMAC BULLETIN NO 10, JULY 1974
MABA
B BECKS
B BECKS, ZEL, KFA, JUELICH
DECTAPE
JANUARY 1974
PDP-11, 16K 16 BIT WORDS OF MEMORY
TYPE 1533A (HOLER)
DOS VOR OR V09, 16K
COMPILE
BASIC
IN-LINE
CAMAC SOFTWARE IS MACRO ASSEMBLER
SINGLE ACTION STATEMENTS

DESCRIPTION =
THIS COMPILER TRANSLATES TESTED (INTERPRETIVE) BASIC
PROGRAMS INTO MACRO-11 SOURCE CODE. MAX TIME IS IMPROVED BY
A FACTOR OF 15 TO 20, EASILY ADAPTABLE TO OTHER CONTROLLERS
(MACROS).
OUTPUT CODE LINKED WITH FLOATING POINT PACKAGE CAN RUN ON
STAND-ALONE MINI-COMPUTERS.

READER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
PUBL, REF, =
NAME/ACRONYM =
OBTAINABLE FROM =
AVAILABLE UN/AS
OPERATIVE DATE =
COMPUTERS =
INTERFACE(S) =
MIN MEMORY SPACE
SOFTWARE TYPE =
LANGUAGES =
INCRP TECHNIQUE
HOST LANGUAGE =
FACILITIES =

REF NO 14,5035
.543
PRECOMPILE FOR IML SUBSET
M. KNEIS
CAMAC BULLETIN NO 10, JUNE 1974, AND GFK
REPORT RFX2121, GFK, 1975 (IN PRESS)
META-II/X
M. KNEIS, IAK II/CYCLUTRON, GFK,
D 7800 KARLSRUHE, POSTFACH 3640
TAPE, CARDS
JULY 1974
IBM/370 (TRANSL.), CDC 3100 (EXECUTION)
IN-MOUSE TYPE
36K BYTES (MAX 86K BYTES)
PRECOMPILE (METACOMPILER SYSTEM)
IML (USER), FORTHAN IV (SYSTEM),
META-II (FOR COMPILER/Writing)
IN-LINE
COMPASS ASSEMBLER (CDC 3100)
SINGLE ACTIONS, MULTIPLE ACTION(MA)
BLUCKTRANSFER(UHL), AND LAM=
CRATE, AND SYSTEM=STATEMENTS

DESCRIPTION =
META-II/X IS A SYSTEM FOR WRITING COMPILED. THE IMPL-
MENTED VERSION OF THE IML PRECOMPILE IS A CROSS-COMPILER
VERSION, I.E. TRANSLATION IS DONE ON AN IBM/370, EXECUTION
ON A CDC 3100 COMPUTER. THE OBJECT CODE FOR PRECOMPILING IS
THE MNEMONIC COMPASS ASSEMBLER (CDC), THEREFORE AN ADDITION-
AL ASSEMBLER STEP IS INVOLVED. WITH META-II/X A PRECOM-
PILE CAN BE WRITTEN AND TESTED IN A FEW DAYS. THE IML SUB-
SET CONTAINS THE DECLARATION (LUCL, LUCU) AND ACTION=STATE-
MENTS (SA, SJU, SJNU, MA, UBL, ALL LAM HANDLING, SYSTEM=
AND CRATE=CONTHULLM= STATEMENTS).
SET CONTAINS THE DECLARATION STATEMENTS LUCL AND LUCU. THE
SUBSET ALSO CONTAIN ACTION STATEMENTS SUCH AS SA, SJU, SJNU,
MA, UBL, ALL LAM HANDLING STATEMENTS, SYSTEM STATEMENTS, AND
CRATE CONTHULLM STATEMENTS.

READER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
PUBL, REF, =
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
SOFTWARE TYPE =
INCRP TECHNIQUE
ENVIRONMENT FOR =
LANGUAGE =

REF NO 14,5036
.544(BASIC)
A PDP-11 BASIC EXTENSION FOR CAMAC
PROGRAMMING
I BALS, E DE AGOSTINO, CNEN, ROME
CAMAC BULLETIN NO 7, JULY 1973
1973
PDP-11
EXECUTIVE SUITE (GEC ELLIOTT)
INTERPRETER
SUBROUTINES IN ASSEMBLY CODE
CAMAC SOFTWARE IS BASIC
BASIC (EXTENDED)

DESCRIPTION =
THE SUBROUTINES WHICH EXTEND THE BASIC INTERPRETER TO CAMAC
ARE CALLED BY AN EXTERNAL FUNCTION STATEMENT, WHERE ADDRESS,
FUNCTION, ETC, ARE TRANSMITTED AS ARGUMENTS. THE STATEMENT
HAS THE FOLLOWING GENERAL FORM =
LET U = EXP (A1,A2, ...,A10)
THE FIRST ARGUMENT SELECTS THE APPROPRIATE SUBROUTINE,
DATALESS, READ, AND WRITE OPERATIONS WITH DIRECT/INDIRECT
ADDRESSING ARE POSSIBLE, ALSO SINGLE OR BLUCK TRANSFERS IN
ADDRESS SCAN, REPEAT OR STOP MODES CAN BE PERFORMED,
THE EXTENSION FEATURES LAM HANDLING.

READER SERVICE
CLASS CODE =
TITLE =
AUTHOR(S) =
PUBL, REF, =
NAME/ACRONYM =
OBTAINABLE FROM =
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
MIN SYSTEM CONFIG
SOFTWARE TYPE =
LANGUAGE =
INCRP TECHNIQUE
CAMAC FACILITIES

REF NO 14,5037
.544(BASIC)
A CAMAC EXTENDED BASIC LANGUAGE
J M SERVENT (SCHLUMBERGER)
PRUC CAMAC SYMPOSIUM, LUXMBG, DEC 1973
CASIC
SCHLUMBERGER (SEE INDEX OF MFWS)
1973
PDP-11, 16K WORDS MEMORY
ICP11 ON JCC11 (SCHLUMBERGER)
TTY
INTERPRETIVE LANGUAGE, EXTENDED
WITH MACRO-INSTRUCTION GENERATION
BASIC (EXTENDED)
IN-LINE CAMAC STATEMENTS
SYMBOLIC DEVICE NAMES, INTERRUPT
HANDLING, RE-ENTRANT.

DESCRIPTION =
STANDARD BASIC IS EXTENDED WITH A SET OF CAMAC RELATED
STATEMENTS. EXECUTION TIME FOR A 100 LINE PROGRAM IS ABOUT
10 SECONDS. DECLARATIVE STATEMENTS ALLOW SYMBOLIC REFERENCE
OF A MODULE. ADDRESS PARAMETERS CAN BE CONSTANTS
OR VARIABLES, EVEN EXPRESSIONS, THUS PROVIDING GREAT
FLEXIBILITY. SEVERAL CONTROL FUNCTIONS ARE IN MACRO=STATE-
MENT FORM, SUCH AS = TST LAM MODULE (SAME AS MODULE(B)),
SOME SYNTAX CHANGES FACILITATES IMPLEMENTATION OF THE SEMAN-
TICS OF IML (SEE .501(IML)). TYPICAL STATEMENTS ARE =
ASSIGN ADDRESS = = STATN(MODULE) = (B,C,N,A)
EXECUTING STATEMENT = =
SINGLE TRANSFER = = SA(F,MODULE,A)
MULTIPLE TRANSFER = = MA(F,MODULE,A)
CONTROL FUNCTION = EXEC MODULE(F)
LAM REG OPERATION = CLR LAM MODULE (MODULE(10))
LAM/INTERRUPT = = UN LAM(MODULE) DU 100

READER SERVICE
CLASS CODE =
TITLE = * * *
AUTHOR(S) =
PUBL. REF. =
NAME/ACRONYM =
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
SOFTWARE TYPE =
INCORP TECHNIQUE
ENVIRONMENT FOR =

REF NO 14,5038
.544(FUCAL)
FUCAL OVERLAY FOR CAMAC DATA
AND COMMAND HANDLING
F MAY, M HALLING, K PETRELCZEK
CAMAC BULLETIN NO 1, JUNE 1971
FOCADAT
1970
PDP-8, 4K OR 8K 12 BIT WORD MEMORY
IN-HOUSE CC & INTERFACE
INTERPRETER (EXTENDED)
CAMAC EXTENSION OF OVERLAY,
IN-LINE CODING OF CAMAC COMMANDS
CAMAC SOFTWARE IS FUCAL

DESCRIPTION =
THE INTERPRETER IS PRIMARILY INTENDED FOR EASILY PROGRAMMED
ON-LINE CAMAC SYSTEMS IN NON-TIME-CRITICAL CONTROL AND DATA
HANDLING APPLICATIONS AND FOR TEST ROUTINES.
THERE ARE 9 CAMAC STATEMENT TYPES COVERING GENERAL CONTROLS
(2, C, I) AND CAMAC COMMANDS WITH/WITHOUT DATA TRANSFER,
THE GENERAL FORM OF A CAMAC STATEMENT IS --
*A CF,C,N,A,P,PB,MM {,L,M,Q}
*WHERE SEVERAL PARAMETERS MAY BE OMITTED,

READER SERVICE
CLASS CODE =
TITLE = * * *
AUTHOR(S) =
VERSION = * * *
MAINTENANCE BY =
OBTAINABLE FROM
AVAILABLE UN/AS
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
MIN SYSTEM CONFIG
SOFTWARE TYPE =
LANGUAGE =
INCORP TECHNIQUE

REF NO 14,5039
.544(BASIC)
B=USER BASIC UNDER DOS WITH
INTERPRETER EXTENDED FOR CAMAC
PFEIFFER, SPICKMAN, CAMLEBACH
001
D P PFEIFFER
D P PFEIFFER, ZAM, KFA, JUELICH
DECTAPE
JANUARY 1974
PDP-11, 16K OF 16 BIT WORD MEMORY
TYPE 153JA (RDWR)
DOS V08 OR V09, 16K
DOS SYSTEM INTERFACE TO CAMAC
BASIC
EXTENSION OF INTERPRETER

DESCRIPTION =
THE B=USER BASIC CAN BE RUN UNDER DOS, A HELP FILE CONTAINS
ALL MODIFICATIONS OF THE 1 TO 8 USER BASIC, NO INTERRUPT
HANDLING, COMMUNICATION BETWEEN THE 8 USERS IS POSSIBLE BY
ONE COMMUNICATION WORD PER USER, EXPANDED ERROR MESSAGE
HANDLING, FILE HANDLING EXTENDED, TIME COMMAND ADDED,

READER SERVICE
CLASS CODE =
TITLE = * * *
AUTHOR(S) =
NAME/ACRONYM =
MAINTENANCE BY =
OBTAINABLE FROM
AVAILABLE UN/AS
OPERATIVE DATE =
COMPUTER =
INTERFACE(S) =
MIN SYSTEM CONFIG
SOFTWARE TYPE =
LANGUAGE =
INCORP TECHNIQUE
CAMAC FACILITIES

REF NO 14,5040
.544
ORACL (TM), AN INTERPRETIVE REAL-
TIME MONITOR WITH CAMAC SUPPORT
L BYARS, R KEYSER (ORTEC INC)
ORACL (TM)
ORTEC
ORTEC (SEE INDEX OF MANUFACTURERS)
PAPER TAPE AND DISK
APRIL 1974
PDP-11, MIN 5K 16 BIT MEMORY
TYPE DC011 (EG&G)
TTY & DC011
INTERPRETER, SYSTEM MONITOR
PDP-11 ASSEMBLER
EMBEDDED CAMAC FEATURES
SINGLE OR MULTIPLE INSTRUCTIONS,
DEMAND HANDLING IS INCLUDED,

DESCRIPTION =
ORACL INTERPRETER IS ARITHMETIC STATEMENTS, PROGRAM CONTROL
STATEMENTS, COMMENTS, I/O STATEMENTS, AND NAME NAME CONTROL
STATEMENTS AND EXECUTES THE DESIRED FUNCTION,

ORACL (TM) IS A TRADE MARK REGISTERED BY UNTEC, INC.

READER SERVICE
CLASS CODE =
TITLE = * * *
AUTHOR(S) =
MAINTENANCE BY =
OBTAINABLE FROM =
OPERATIVE DATE =
AVAILABLE UN/AS =
COMPUTER =
INTERFACE(S) =
SOFTWARE TYPE =
OTHER REMARKS

REF NO 14,5041
.544
GENERAL PURPOSE I/O INTERFACE SOFTWARE
F MURM, SEN ELECTRONIQUE
SEN
SEN (SEE INDEX OF MANUFACTURERS)
MAY 1975
DISK
NOVA SERIES (DATA GENERAL)
ANY (IRRESPECTIVE OF MAKE)
INTERPRETER
FULLY DOS/SOS COMPATIBLE

DESCRIPTION =

. 55 Support Software II

READER SERVICE
CLASS CODE =
TITLE = * * *
AUTHOR(S) =
PUBL. REF. =
NAME/ACRONYM =
OPERATIVE DATE =
COMPUTER =
SOFTWARE TYPE =

REF NO 14,5042
.555(FUCAL/PAL)
A FUCAL INTERRUPT HANDLER FOR CAMAC
F MAY, M MARSHIK, M HALLING
CAMAC BULLETIN NO 6, MARCH 1973
FUCALINT
1971
PDP-8
INTERRUPT HANDLER (SYSTEM PROGRAM)

DESCRIPTION =
FUCALINT IS A GENERAL PURPOSE SYSTEM PROGRAM, ADAPTABLE FOR
SPECIAL USE, UP TO 3 CRATES WITH 24 INTERRUPTS EACH CAN BE
SERVICED, ONE PROGRAM LINE IN FUCAL IS RESERVED FOR EACH
INTERRUPT, SHORT ROUTINES CAN BE TYPED INTO THESE LINES
SERVICING THE ASSOCIATED INTERRUPTS, ALTERNATIVELY A FUCAL
SUBROUTINE CAN BE USED, CURRENT LINE IN THE BACKGROUND
PROGRAM WILL BE FINISHED BEFORE JUMPING TO INTERRUPT ROUTINE
AND RETURNS TO NEXT LINE IN THE BACKGROUND PROGRAM AFTER
SERVICING,

.57 Test Routines

<p>READER SERVICE CLASS CODE = TITLE = * * *</p> <p>OBTAINABLE FROM OTHER REMARKS</p>	<p>REF NO 14,5043 .57 TEST PROGRAMS FOR SYSTEMS, BRANCH DRIVER & MODULES BI RA SYSTEMS (SEE INDEX OF MFMS) FOR BRANCH DRIVER MOD=11, SYSTEM TEST MODULE 0102, AND DATA MODULES</p>	<p>DESCRIPTION = A SET OF THREE DIAGNOSTIC PROGRAMS ARE SUPPLIED WITH THE MOD=11 MICROPROGRAMMED BRANCH DRIVER, TESTS OF MEMORY, FILE REGISTERS, INSTRUCTION SET, DMA TRANSFERS, INTERRUPTS ETC. A COMPLETE SYSTEM TEST IS SUPPLIED WITH 0102. A CAMAC TEST ROUTINE IS SUPPLIED FOR CAMAC MODULE TESTING FROM THE TELETYPE, NO ASSEMBLY LANGUAGE KNOWLEDGE REQUIRED.</p>
<p>READER SERVICE CLASS CODE = TITLE = * * * AUTHOR(S) = AVAILABLE UN/AS OPERATIVE DATE = COMPUTER = INTERFACE(S) = SOFTWARE TYPE =</p>	<p>REF NO 14,5044 .573 CAMAC TEST PROGRAM DR. B. MERTENS, IKP, KFA, JUELICH PAPER TAPE, ASCII CODE 1971 PDP=11, 16K OF 16 BIT WORDS MEMORY TYPE 2200 (BUREAU) TEST ROUTINES, STAND-ALONE PROGRAMS</p>	<p>DESCRIPTION = STAND-ALONE PROGRAMS TEST SOME FUNCTIONS OF THE BUREAU TYPE 2200 INTERFACE, THE LATE CONTROLLER AND TWO IN-MOUSE MODULES (C01 & C02). LRRUN MESSAGES ARE OUTPUT IF THERE ARE HARDWARE FAILURES.</p>
<p>READER SERVICE CLASS CODE = TITLE = * * * AUTHOR(S) = OBTAINABLE FROM OPERATIVE DATE = COMPUTER = INTERFACE(S) = SOFTWARE TYPE =</p>	<p>REF NO 14,5045 .573 3911A TEST CAMAC L. A. KLAISNER KINETIC SYSTEMS (SEE INDEX OF MFMS) 1973 PDP=11, 4K OF CORE MEMORY REQUIRED TYPE 3911A (KINETIC) TEST ROUTINE</p>	<p>DESCRIPTION = A STAND-ALONE PROGRAM FOR EXERCISING A CAMAC SYSTEM FROM A TELETYPE. IT SUPPORTS UP TO 8 CHATES WITH MODEL 3911A UNIBUS *) CHATE CONTROLLERS. A FUNCTION MAY BE EXECUTED ONCE OR REPETITIVELY.</p> <p>*) UNIBUS IS A TRADE MARK OF DIGITAL EQUIPMENT CORP.</p>
<p>READER SERVICE CLASS CODE = TITLE = * * * OBTAINABLE FROM OPERATIVE DATE = INTERFACE(S) = COMPUTER = SOFTWARE TYPE =</p>	<p>REF NO 14,5046 .573 TEST CAMAC KINETIC SYSTEMS (SEE INDEX OF MFMS) 1972 TYPE KS0011 (KINETIC) PDP=11, 4K OF CORE REQUIRED TEST ROUTINE</p>	<p>DESCRIPTION = A STAND-ALONE PROGRAM FOR EXERCISING A CAMAC SYSTEM FROM A TELETYPE. IT SUPPORTS ONE BRANCH DRIVER WITH UP TO 7 CHATES. A FUNCTION MAY BE EXECUTED ONCE OR REPETITIVELY.</p>
<p>READER SERVICE CLASS CODE = TITLE = * * * OBTAINABLE FROM OPERATIVE DATE = COMPUTER = INTERFACE(S) = SOFTWARE TYPE = LANGUAGE =</p>	<p>REF NO 14,5047 .573 PDP=11 INTERFACE TEST PROGRAM GEC=ELLIOTT (SEE INDEX OF MFMS) 1974 PDP=11 PDP=11 EXECUTIVE SUITE/GEC=ELLIOTT TEST ROUTINE PAL=11 ASSEMBLER</p>	<p>DESCRIPTION = THIS IS A STAND-ALONE PROGRAM USED IN CHECKING THE EXECUTIVE SUITE. A MODULAR PDP=11 - CAMAC INTERFACE, DIAGNOSTIC MESSAGES ARE ISSUED.</p>
<p>READER SERVICE CLASS CODE = TITLE = * * *</p> <p>OBTAINABLE FROM = COMPUTER = INTERFACE(S) = SOFTWARE TYPE = WIN SYSTEM CONFIG</p>	<p>REF NO 14,5048 .573 TEST PROGRAMS FOR BRANCH DRIVER AND SYSTEM WITH MODULE 0102 AND TYPE A BI RA SYSTEMS (SEE INDEX OF MFMS) PRIME COMPUTER 1200 (BI RA SYSTEMS) DIAGNOSTIC PROGRAMS BRANCH DRIVER 1200, 0102 CAMAC TEST MODULE/DATAWAY DISPLAY</p>	<p>DESCRIPTION = A SET OF DIAGNOSTIC PROGRAMS ARE SUPPLIED WITH THE MODEL 1200 PRIME COMPUTER BRANCH DRIVER. A COMPLETE SYSTEM TEST IS SUPPLIED, BUT REQUIRES MODEL 0102 TEST MODULE.</p>

APPENDIX II

SPACELAB SOFTWARE DESCRIPTION

(Reproduced from "Spacelab Payload Accommodations Handbook,"
ESA, May 1976)

4.5 Software

The Spacelab Computer Software comprises the software used for Spacelab during software developments, integration, testing, and operation. This includes subsystem testing, integration, checkout, on-board data handling for subsystems, on-board data handling support for experiments, and checkout for the CDMS portion of the experiment interfaces. Also included is certain support software used in the generation and validation of software and for the off-line reduction and analysis of checkout data.

Software especially dedicated to experiments is not included in the Spacelab computer software.

The Spacelab computer software is made up of sets, each of which is the assembly of software, used for a particular phase of the Spacelab program, with a specific computer system (experiment computer, S/S computer, EGSE, or Software Development Facility).

A set is made up of a number of packages.

A package consists of a group of software modules which are used together to perform some clearly definable functions.

Fig 4.5-1 thru-3 give an overview about SL computer software and the interrelationship between packages.

The Spacelab computer software is designed in a modular way in order to allow for good testability and maximum use of common functional units. Thus commonality can be achieved between the experiment and subsystem computers concerning the operating system and general facilities, such as operator interface, monitoring, fault isolation, subroutine library, etc.

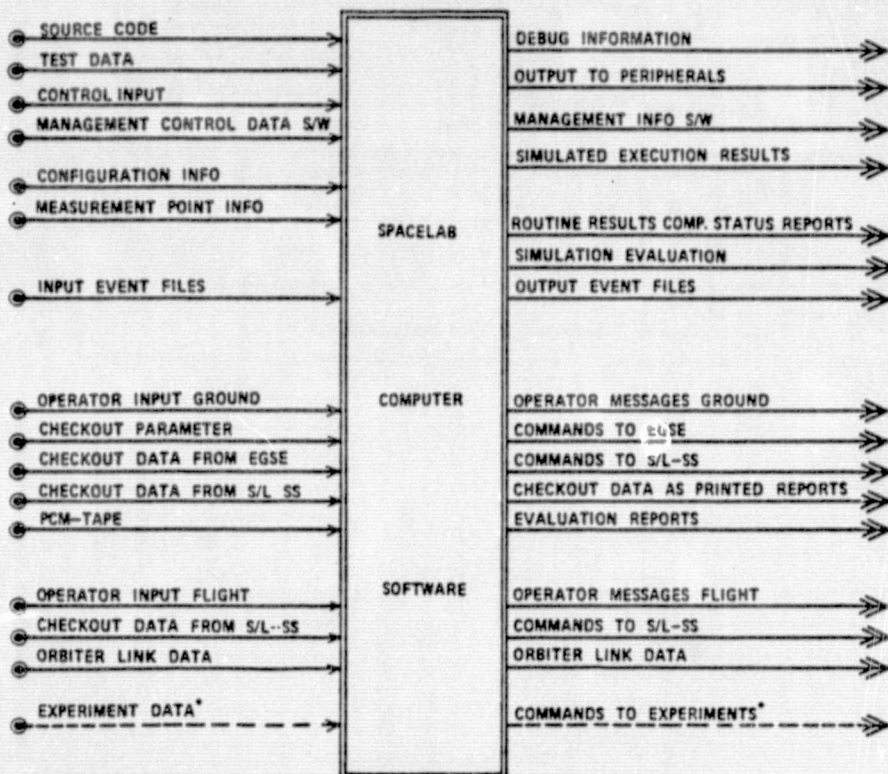
Packages relevant for the experimenter are:

- CDMS Computer Operating System Packages

This package consists of the subsystem computer operating system (SCOS) and the experiment computer operating system (ECOS). For details see 4.5.1.1 -

- Support Software Packages.

The experiment application software packages running in the experiment computer under the ECOS will be supplied by the experimenter and/or the payload integrator.



*NO RESPONSIBILITY OF
SPACELAB CONTRACTOR

Figure 4.5 - 1

Spacelab Computer Software

ORIGINAL PAGE IS
OF POOR QUALITY

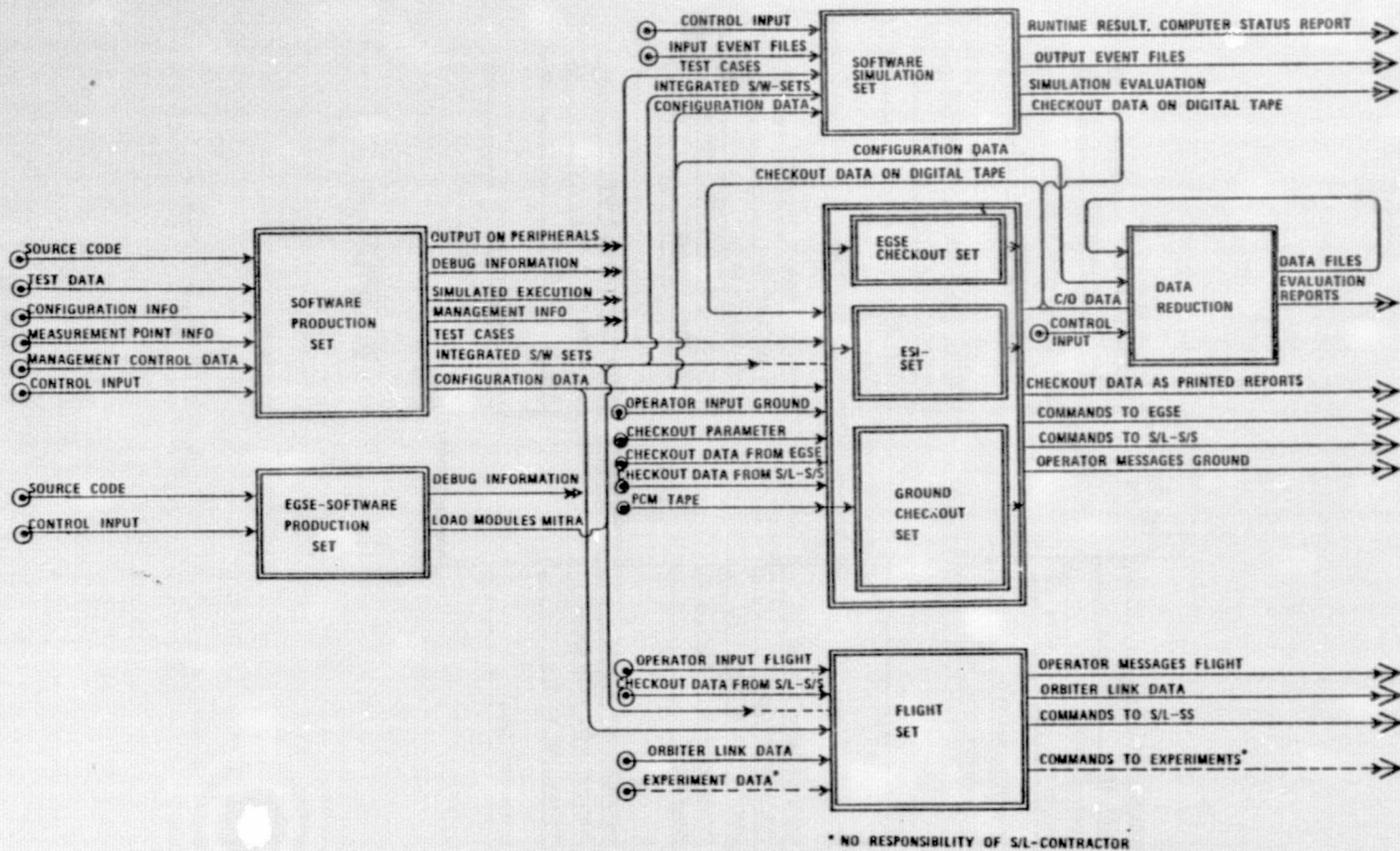
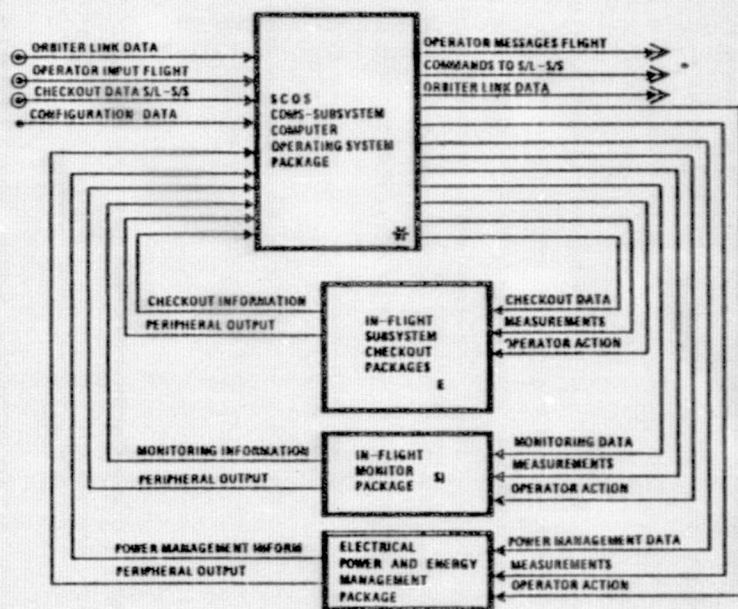


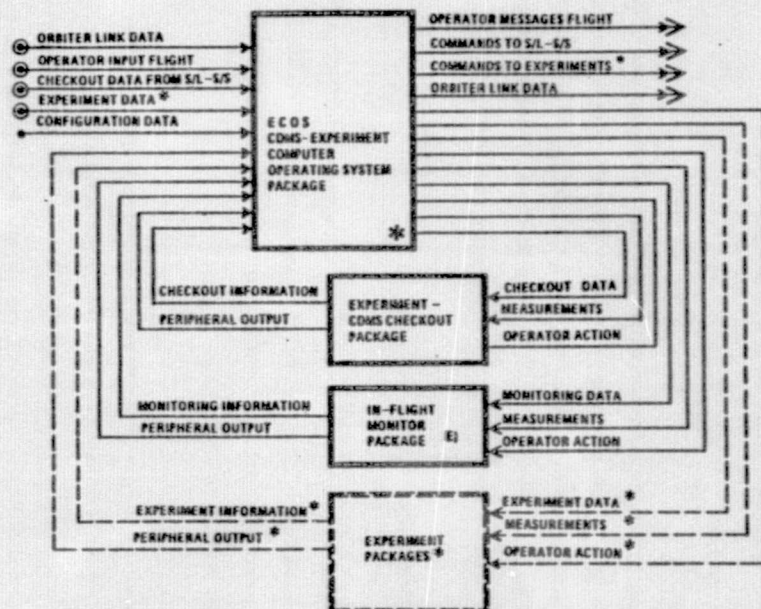
Figure 4.5 - 2 Sets of Spacelab Computer Software



E: THE FOLLOWING IN-FLIGHT SUBSYSTEM CHECKOUT PACKAGES EXIST:

- EPS - C/O
- ECS - C/O
- S/S COMS C/O

*REMARK: FOR THE SAKE OF CLARITY THE CHECKOUT-INTERPRETER IS CONSOLIDATED TRANSPARENT. THEREFORE ALL INPUTS/OUTPUTS THE CHECKOUT APPLICATION SOFTWARE HAS TO DEAL WITH THROUGH THE INTERPRETER ARE ROUTED TO/FROM THE CHECKOUT APPLICATION SOFTWARE PACKAGES



S: MEANS COMS - S/S - COMPUTER RESIDING MODULES OF PACKAGE

E: MEANS COMS - EXP - COMPUTER RESIDING MODULES OF PACKAGE

* NO RESPONSIBILITY OF SPACELAB CONTRACTOR

Figure 4.5 - 3 Flight Software Set

ORIGINAL PAGE IS
OF POOR QUALITY

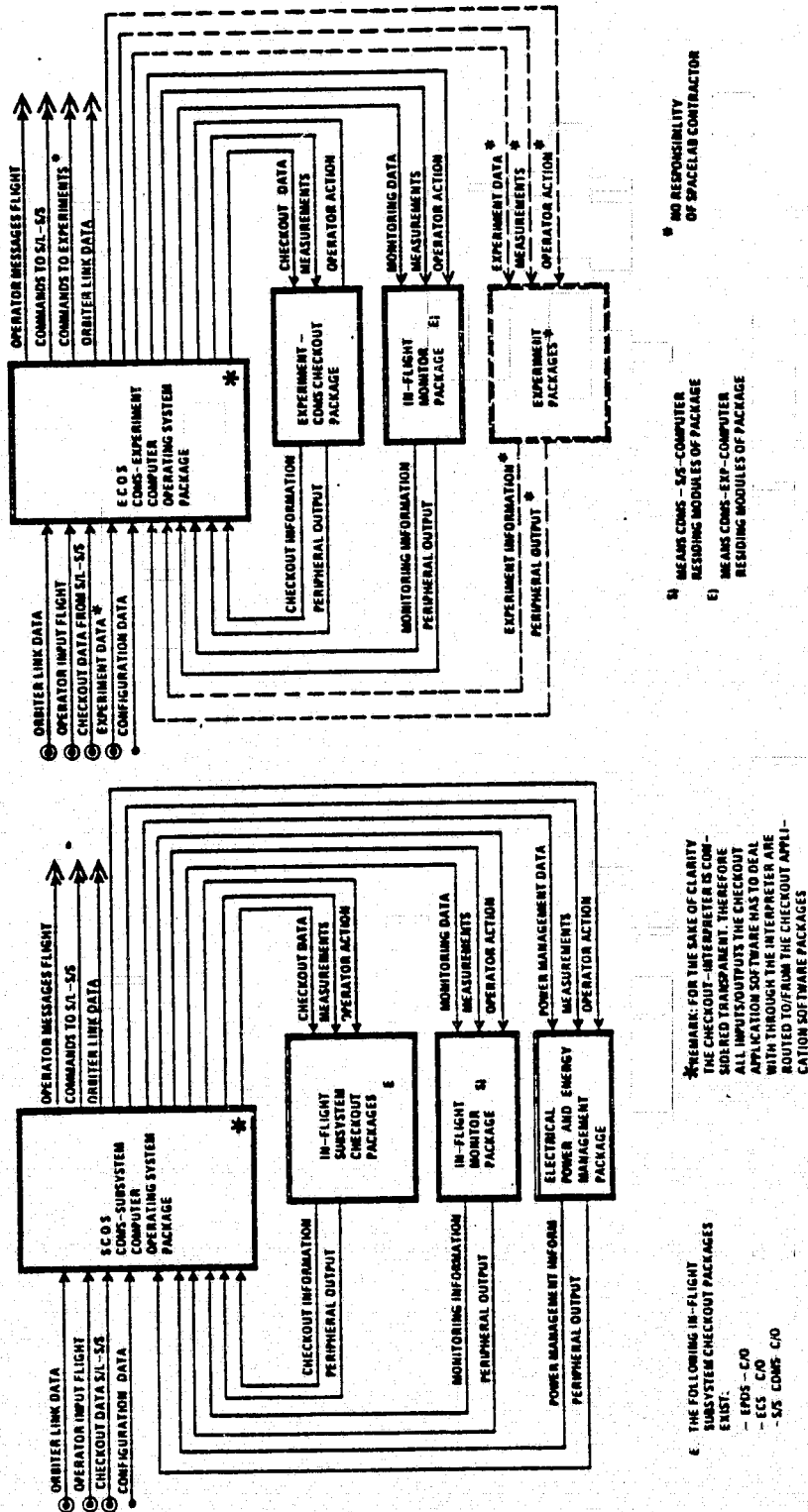


Figure 4.5-3 Flight Software Set

ORIGINAL PAGE IS
OF POOR QUALITY

4.5.1 Spacelab Software Environment

The experimenter - when linking up his experiment software with the Spacelab computer software - has to deal with the CDMS computer operating system package running in the experiment computer (ECOS), and with those modules of the flight application software packages (FLAP) which are also applicable to the experiments and which can be regarded as facilities available for applications. Furthermore, means are provided to support the experimenter in compiling, testing and integrating his software.

4.5.1.1 CDMS Computer Operating System

The CDMS computer operating system is at present the same for the subsystem (SCOS) and the experiment (ECOS) computers. However, because of the requirement that the experiment computer operating system accommodates a variety of experiment applications the ECOS may eventually grow to include greater capability in the areas of control and data processing.

The ECOS allows for asynchronous as well as synchronous tasks to be performed. The executive performs initialization, scheduling and termination of tasks. It assures time scheduling, loading of tasks including overlay and memory allocation to them, management of the various data tables in the data base for program control and housekeeping. It controls the allocation of the computer peripherals, such as memory, keyboards, data display unit, telemetry channels, and data bus. The executive allows for initialization of the computer system and for convenient recovery after system failure. The executive includes a computer self check which is executed periodically, providing a message in case of failure.

The input/output functions provide all services necessary to operate the remote acquisition units (RAU's para. 4.4.2.1). They format and transmit data to the CRTs for presentation to the crew and experimenters, receive and process external event messages based upon usage of the keyboard and inputs from experimenters. They permit communication with the Orbiter for reception of commands, state vector and timing data. They perform the transmission of data to the Orbiter for inclusion in downlink telemetry. They check the status of the peripherals (parity checking, data ready bits, data available bits as applicable).

The functions summarized as general facilities are functions common to all or most of the application programs and include services such as converting of raw data into engineering units, library of mathematical functions, etc.

The ECOS is able to monitor experiments, and to perform limit checking and calibration of data for display.

The ECOS is considered to be core resident with the exception of display formatting routines which will be stored on the mass memory (ref. para. 4.4).

The size of the ECOS is TBD.

Average operating system overhead is estimated to be 5 % of CPU time. Reaction time of external events is estimated to be 100 μ sec maximum.

The S/W - S/W interface between the ECOS and experiment application packages is managed by supervisor calls and data tables.

The S/W - H/W interface between the ECOS and the peripheral hardware is handled via drivers in the ECOS which perform activation, status check data transfer and termination on the peripheral.

A keyboard language for communication between operator and experiment computer will be provided, thus the ECOS provides the interface between the operator and the computer system.

The functions involved are calling for data display and computer status display, initiating and termination of experiment modules at predefined points, interpretation of keyboard commands and changes to experiment modules.

The ECOS will be capable of displaying on CRT structures representing all the groups of data which may be selected for display. In addition, the capability will be provided to generate and display on-line a specific list of data on operator request.

4.5.1.2 Facilities Available for Application

The experiment application software for the experiment computer is the software executed by the ECOS and consists only of the monitor and fault isolation module for the experiment portion of the CDMS. All experiment related software packages to be loaded in the experiment computer are produced by the experimenters.

Only some of the modules of the Spacelab flight application software (FLAP) can also be used by the experimenter.

Within the FLAP there are modules available for management of electrical power and energy which make the respective information available for the experimenter on CRT by request via a keyboard entry to the subsystem computer. In addition, it will be possible to update in predefined areas of memory values and limits per telecommand and/or keyboard.

ORIGINAL PAGE IS
OF POOR QUALITY

4.5.1.3 Software Integration

For integration of his experiment application software, the experimenter will be supplied with the following software (para. 4.5.2):

- CDMS - simulator
This software will simulate on a host computer the CDMS environment
- Interpretive computer simulator (ICS)
This software simulates the Mitra 125 S/MS on a host computer
- Experiment computer operating system (ECOS)

The CDMS environment simulator and the ICS can be integrated in order to simulate the complete CDMS on the host computer.

4.5.2 Software Development Aids

This software is part of the support software packages to be used for the effective development and maintenance of all Spacelab software, i.e. not only for the experiment software but also for operating systems, ground checkout packages and the flight application packages.

This means the experimenter, in developing his experiment software, shall utilize the facilities provided as far as possible. Experiment software shall be written in one of the languages explained in para.

4.5.2.2 which are available with the host software system (see para. 4.5.2.1). For debugging the simulator software shall be employed.

4.5.2.1 Host Software System

The host software system comprises all that support software necessary for the development of all experiment software and executed on a host computer (IBM/370). The following items will be available.

- HAL/S - 370 Compiler System
This compiler system can be used to test programs written in HAL/S on an IBM 370. The compiler will compile HAL/S statements into code executable on an IBM 370 computer. The system also includes an execution monitor under which the compiled code can be executed.
- HAL/S - CII Compiler
This compiler will compile HAL/S statements into code executable on a Mitra 125S/MS computer. The compiler itself will run on an IBM 370.

- **GOAL Compiler**
The GOAL compiler will compile GOAL checkout statements into interpretative code. The interpretative code can be executed by an interpreter running on a Mitra 125S/MS computer. The compiler itself will run on an IBM 370.
- **Interpretative Computer Simulator (ICS)**
The ICS will simulate the Mitra 125 S/MS. This simulator will execute on an IBM 370.
- **Mitra 125 S/MS Macro Assembler (MAS)**
Two versions of the assembler will be available. One will execute on IBM 370 and one will execute on the Mitra 125 itself. Code generated by either one can be processed by the EDL (see below).
- **Mitra 125 S/MS Linkage Editor (EDL)**
Two versions of the EDL will be available. One will execute on IBM 370 and one will execute on the Mitra 125 itself. Code generated by either one can be processed by the preloader.
- **I/O Box and Peripheral Simulator (IOBPS)**
This simulator will simulate the reactions of all CDMS hardware (except the computer) with respect to computer input/output and outside events. The IOBPS can work together with the ICS. It will execute on an IBM 370.

4.5.2.2 Programming Languages

- **HAL/S**
Experiment software may be written in the programming language HAL/S. This is a real time programming language which allows the scheduling and synchronization of programs. The language also allows the manipulation of vectors and matrices and data structures in a simple manner.

A wide range of mathematical functions is available with HAL/S.
- Experiment software may be written in C II MITRA 125 S/MS assembler language.
- Checkout software for experiments may be written in the checkout language GOAL. This language is oriented towards the convenient specification of checkout procedures by scientists and engineers.

ORIGINAL PAGE IS
OF POOR QUALITY

4.5.3 Software Development Guidelines

Software development guidelines and standards, as well as procedures for the technical management, will be provided within the Software Standards Manual (Doc. No. MA-ER-0001).

There are two main topics: One covers the part of technical management such as verification (reviews and acceptance) and configuration control. The other specifies the necessary guidelines and standards to be followed during software development (design, implementation, test and documentation) to satisfy the requirements of software control.

As far as the user's interaction with NASA/ESA is concerned and to enable NASA/ESA to control and integrate the experiment software, the user will also have to follow some of the corresponding procedures and guidelines within the Software Standards Manual.

The relevant topics will be referenced in a manual of guidelines for experiment software. Additional guidelines, e.g. safety requirements, constraints on size and frequency and overall memory requirements, will be included.